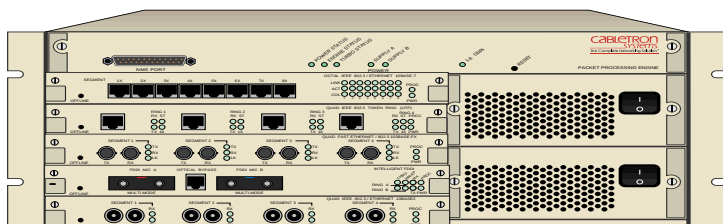


ATX USER'S GUIDE



NOTICE

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.

© Copyright 1997 by Cabletron Systems, Inc., P.O. Box 5005, Rochester, NH 03866-5005
All Rights Reserved
Printed in the United States of America

Order Number: 9031871-02 April 1997

FCC NOTICE

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment uses, generates, and can radiate radio frequency energy and if not installed in accordance with the operator's manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause interference in which case the user will be required to correct the interference at his own expense.

WARNING: Changes or modifications made to this device which are not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

DOC NOTICE

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

VCCI NOTICE

This equipment is in the 1st Class Category (information equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council for Interference by Information Technology Equipment (VCCI) aimed at preventing radio interference in commercial and/or industrial areas.

Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc.

Read the instructions for correct handling.

この装置は、第一種情報装置（商工業地域において使用されるべき情報装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会（VCCI）基準に適合しております。

従って、住宅地域またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。

取扱説明書に従って正しい取り扱いをして下さい。

EXCLUSION OF WARRANTY AND DISCLAIMER OF LIABILITY

1. **EXCLUSION OF WARRANTY.** Except as may be specifically provided by Cabletron in writing, Cabletron makes no warranty, expressed or implied, concerning the Program (including its documentation and media).

CABLETRON DISCLAIMS ALL WARRANTIES, OTHER THAN THOSE SUPPLIED TO YOU BY CABLETRON IN WRITING, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PROGRAM, THE ACCOMPANYING WRITTEN MATERIALS, AND ANY ACCOMPANYING HARDWARE.

2. **NO LIABILITY FOR CONSEQUENTIAL DAMAGES.** IN NO EVENT SHALL CABLETRON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR RELIANCE DAMAGES, OR OTHER LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THIS CABLETRON PRODUCT, EVEN IF CABLETRON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR ON THE DURATION OR LIMITATION OF IMPLIED WARRANTIES, IN SOME INSTANCES THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

CABLETRON SYSTEMS, INC. PROGRAM LICENSE AGREEMENT

IMPORTANT: Before utilizing this product, carefully read this License Agreement.

This document is an agreement between you, the end user, and Cabletron Systems, Inc. ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software program (the "Program") contained in this package. The Program may be contained in firmware, chips or other media. BY UTILIZING THE ENCLOSED PRODUCT, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE LICENSE AND THE LIMITATION OF WARRANTY AND DISCLAIMER OF LIABILITY. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

CABLETRON SOFTWARE PROGRAM LICENSE

1. **LICENSE.** You have the right to use only the one (1) copy of the Program provided in this package subject to the terms and conditions of this License Agreement.

You may not copy, reproduce or transmit any part of the Program except as permitted by the Copyright Act of the United States or as authorized in writing by Cabletron.

2. **OTHER RESTRICTIONS.** You may not reverse engineer, decompile, or disassemble the Program.
3. **APPLICABLE LAW.** This License Agreement shall be interpreted and governed under the laws and in the state and federal courts of New Hampshire. You accept the personal jurisdiction and venue of the New Hampshire courts.

DECLARATION OF CONFORMITY

Application of Council Directive(s): **89/336/EEC**
73/23/EEC

Manufacturer's Name: **Cabletron Systems, Inc.**

Manufacturer's Address: **35 Industrial Way**
PO Box 5005
Rochester, NH 03867

European Representative Name: **Mr. J. Solari**

European Representative Address: **Cabletron Systems Limited**
Nexus House, Newbury Business Park
London Road, Newbury
Berkshire RG13 2PZ, England

Conformance to Directive(s)/Product Standards: **EC Directive 89/336/EEC**
EC Directive 73/23/EEC
EN 55022
EN 50082-1
EN 60950

Equipment Type/Environment: **Networking Equipment, for use in a**
Commercial or Light Industrial
Environment.

We the undersigned, hereby declare, under our sole responsibility, that the equipment packaged with this notice conforms to the above directives.

Manufacturer
Mr. Ronald Fotino
Full Name
Principal Compliance Engineer
Title
Rochester, NH, USA
Location

Legal Representative in Europe
Mr. J. Solari
Full Name
Managing Director - E.M.E.A.
Title
Newbury, Berkshire, England
Location

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 USING THIS MANUAL	1-1
1.2 DOCUMENT CONVENTIONS	1-3
1.3 RELATED DOCUMENTATION	1-4
1.4 GETTING HELP	1-5
1.5 ATX ARCHITECTURE	1-6
1.6 ATX FEATURES	1-6
1.6.1 Netbios Name Caching	1-8
1.6.2 ATX Local and Remote Port Mirroring	1-9
1.6.3 IPX with Token Ring Source Routing	1-10
1.6.4 Event Logging on the ATX	1-10
1.6.5 ATX LAN Switch Workgroups	1-11
1.6.6 ATX Packet Processing Engine	1-11
1.6.7 Input/output Modules	1-12
1.6.8 Power Supply	1-13
1.7 BRIDGING FUNCTIONS	1-13
1.7.1 Transparent Bridging	1-16
1.7.2 Source Route Translational Bridging	1-16
1.7.3 Source Routing Bridging	1-17
1.7.4 Source Routing Transparent Bridging	1-19
1.7.5 Translation	1-20
1.8 ROUTING FUNCTIONS	1-21
1.8.1 IP Routing	1-22
Routing Information Protocol (RIP)	1-22
Address Resolution Protocol (ARP)	1-22
Reverse Address Resolution Protocol (RARP)	1-23
Proxy ARP	1-23
BOOTP	1-23
IPM	1-23
1.8.2 Multiple IP Networks Per Port	1-24
1.8.3 IP Multicast Routing	1-26
1.8.4 IP Routing Over Source Routing	1-29
1.8.5 Configuring IP Routing Over Source Routing	1-32
1.8.6 IPX Routing	1-32
Routing Information Protocol (RIP)	1-33
Service Advertising Protocol (SAP)	1-33
IPX Routing Over Source Route	1-33

- 1.8.7 Appletalk Routing.....1-34
 - AppleTalk addressing.....1-34
 - AppleTalk zones.....1-34
 - How a Macintosh learns its address.....1-35
 - How a router learns its address 1-35
 - Seed Routers 1-36
- 1.9 TRUNKING.....1-37
 - Trunk Groups.....1-38
- 1.10 LOCAL CONSOLE MANAGER.....1-39
 - 1.10.1 Command Syntax Conventions 1-40
 - 1.10.2 Basic LCM Commands.....1-41

CHAPTER 2 INSTALLING AND CONNECTING TO THE NETWORK

- 2.1 ATX FRONT PANEL 2-1
- 2.2 MOUNTING THE ATX 2-3
- 2.3 CONNECTING THE POWER SUPPLY.....2-4
 - 2.3.1 Checking the Power-up Sequence 2-5
 - Power-up Diagnostics Sequence.....2-6
 - Troubleshooting the Power-up Sequence 2-7
 - Replacing the Power Supply 2-8
- 2.4 CONNECTING THE LOCAL CONSOLE MANAGER 2-10

CHAPTER 3 CONFIGURING

- 3.1 CONFIGURING BRIDGING.....3-1
 - 3.1.1 Enabling Bridging Functions.....3-3
 - 3.1.2 Displaying Bridging Functions 3-4
 - 3.1.3 Disabling Bridging.....3-5
- 3.2 CONFIGURING IP ROUTING 3-5
 - 3.2.1 Assigning an IP Address 3-5
 - 3.2.2 Deleting an IP Address.....3-6
 - 3.2.3 Changing a Subnet Mask 3-7
 - 3.2.4 Displaying IP Addresses 3-7
 - 3.2.5 Enabling IP Routing Functions 3-8
 - 3.2.6 Adding an IP Address to a Port 3-9
 - 3.2.7 Deleting an IP Address From a Port.....3-10
 - 3.2.8 Clearing All IP Addresses From a Port 3-10
 - 3.2.9 IP Multicast Routing LCM Commands 3-11
 - 3.2.10 Displaying IP Routing Functions.....3-12

3.2.11	Disabling Routing Functions.....	3-12
3.3	CONFIGURING IPX ROUTING.....	3-12
3.3.1	Assigning an IPX Address	3-13
3.3.2	Displaying IPX Addresses	3-13
3.3.3	Enabling IPX Routing Functions	3-14
3.3.4	Displaying IPX Routing Functions.....	3-15
3.3.5	Disabling IPX Routing.....	3-15
3.4	CONFIGURING APPLETALK ROUTING.....	3-15
3.4.1	Enabling AppleTalk Routing	3-16
3.4.2	Displaying AppleTalk Routing Functions	3-16
3.4.3	Disabling AppleTalk Routing.....	3-17
3.4.4	Assigning a Network Number.....	3-17
3.4.5	Displaying the Network Number	3-19
3.4.6	Adding a Zone Name.....	3-19
3.4.7	Displaying a Zone Name	3-20
3.5	CONFIGURING TRUNKING	3-20
3.5.1	Enabling Trunking	3-21
3.5.2	Disabling Trunking	3-22
3.6	CONFIGURING MULTICAST STORM PROTECTION	3-22
3.7	MODIFYING MIB VARIABLES.....	3-23
3.7.1	System Contact.....	3-23
3.7.2	System Name.....	3-24
3.7.3	System Location	3-24
3.7.4	Authentication Password.....	3-24
Set Password	3-24	
Get Password	3-25	
Aging Parameter	3-25	
Traps (acknowledge).....	3-25	
Configuration Alarm Dynamic	3-26	
3.8	CONFIGURING NETBIOS NAME CACHING	3-26
3.9	VIRTUAL WORKGROUP LCM COMMANDS.....	3-27
3.10	CLASSIFICATION.....	3-28
3.10.1	Workgroup of Type ALL.....	3-28
3.10.2	Workgroup of Type IP	3-29
3.10.3	Workgroup of Type IPX	3-31
3.10.4	Same Port in Multiple Workgroups	3-33
3.10.5	Workgroup to Workgroup Communication	3-34
3.11	LOCAL AND REMOTE PORT MIRRORING COMMANDS	3-35
3.11.1	Types of Media and Framing.....	3-36
3.11.2	Packet Capturing and Mirroring	3-37
3.11.3	Mirrored Filters	3-38

- 3.11.4 Example #1: LOCAL Port Mirroring 3-38
- 3.11.5 Example #2: REMOTE Port Mirroring.....3-39
- 3.12 IPX ROUTING OVER SOURCE ROUTE COMMANDS.....3-40
- 3.13 PING COMMANDS 3-40
- 3.14 TRACE ROUTE COMMANDS.....3-40
- 3.15 EVENT LOGGING COMMANDS.....3-41
 - 3.15.1 eventfilter 3-41
 - 3.15.2 eventtrap 3-42
 - 3.15.3 eventdisplay 3-42
- 3.16 CONFIGURING SOURCE ROUTE TRANSLATIONAL BRIDGING
(RIF CACHING).....3-42
 - 3.16.1 Managing SRTB.....3-43
 - 3.16.2 SRTB Usage in the ATX 3-44

CHAPTER 4 MONITORING AND MANAGING THE ATX

- 4.1 MONITORING STATISTICS.....4-1
 - 4.1.1 General Status and Statistics 4-4
 - 4.1.2 IP Status and Statistics.....4-4
 - 4.1.3 ICMP Status and Statistics 4-6
 - 4.1.4 UDP Status and Statistics.....4-8
 - 4.1.5 SNMP Status and Statistics.....4-9
 - 4.1.6 Spanning Tree Status and Statistics 4-10
- 4.2 MODULE STATUS AND STATISTICS.....4-11
 - 4.2.1 End-node Status and Statistics 4-11
 - 4.2.2 Traffic Analysis Statistics.....4-12
- 4.3 MONITORING STATUS.....4-13
 - 4.3.1 Displaying Status 4-13
 - 4.3.2 Displaying MAC Addresses 4-15
 - 4.3.3 Displaying Manufacturing Information 4-17
- 4.4 MANAGING YOUR ATX 4-17
 - 4.4.1 Disabling a Port.....4-18
 - 4.4.2 Enabling a Port 4-18
 - 4.4.3 Taking a Module Offline 4-19
 - 4.4.4 Bringing a Module Online 4-19
 - 4.4.5 Setting The Baud Rate 4-20
 - 4.4.6 Displaying The Baud Rate 4-20
 - 4.4.7 Assigning a Community Name 4-20

CHAPTER 5 FILTERS

5.1	FILTERING AND PERFORMANCE CONSIDERATIONS	5-2
5.2	USING FILTERS FOR SECURITY PURPOSES	5-2
5.3	USING FILTERS TO IMPROVE PERFORMANCE	5-3
5.4	ADDRESS TABLE FILTERS	5-4
5.4.1	Destination Address Filter	5-5
5.4.2	Source Address Filter	5-5
5.4.3	Combination Address Filters	5-6
5.4.4	Source Address Multicast Filter	5-6
5.5	COMBINATION PORT FILTERS	5-7
5.5.1	Configurable Fields	5-8
	Type	5-9
	Source Range	5-9
	Source Range Start	5-9
	Source Range End	5-9
	Source Range Mask	5-10
	Destination Range	5-10
	Destination Range Start	5-10
	Destination Range End	5-10
	Destination Range Mask	5-10
	Port/Group Match	5-10
	Port/Group #	5-10
	Protocol Match	5-11
	Protocol Type	5-11
	Field Match	5-11
	Field Origin	5-12
	Field Offset	5-12
	Field Value	5-12
	Field Mask	5-12
	Threshold Time	5-12
	Threshold	5-13
	Filter Index	5-13
	Combination Port Filter Options	5-13
	Pseudo Filter Option	5-13
	Linking Combination Port Filters	5-14
5.6	ADDING A FILTER	5-14
5.7	MODIFYING A FILTER	5-18
5.8	DELETING A FILTER	5-18
5.9	DISPLAYING A FILTER	5-19
5.10	FILTERING APPLICATION EXAMPLES	5-19

- 5.10.1 Filtering for Security Purposes.....5-20
 - Example 1 — Blocking access to a network segment 5-20
 - Example 2 — Blocking access to specific stations 5-22
 - Example 3 — Restricting access to authorized users.....5-25
 - Example 4 — Filtering by vendor ID 5-26
 - Example 5 — Configuring a firewall filter
to control multicasts 5-27

CHAPTER 6 TRAPS

- 6.1 GENERIC SNMP TRAPS.....6-1
- 6.2 ATX UNIQUE TRAP IDS.....6-3

CHAPTER 7 DIAGNOSTICS AND TROUBLESHOOTING

- 7.1 DIAGNOSTICS OVERVIEW 7-1
- 7.2 POWER-UP DIAGNOSTICS 7-1
 - 7.2.1 Power-up LED Sequence 7-2
 - 7.2.2 Specific Power-up Tests.....7-3
 - 7.2.3 Software Checksum Comparison 7-4
 - 7.2.4 Power-up Diagnostics Results.....7-4
 - 7.2.5 Responses to Failures at Power-up.....7-4
 - Failure Indicators 7-5
 - NMS Failure Traps 7-5
- 7.3 DIAGNOSTICS WHILE ATX IS OPERATIONAL 7-5
 - 7.3.1 Loopback Tests 7-5
 - 7.3.2 Diagnostic Results.....7-6
- 7.4 STATUS AND ACTIVITY LEDS.....7-6
- 7.5 TROUBLESHOOTING.....7-8
 - 7.5.1 ATX Does Not Power Up 7-9
 - 7.5.2 Module Status LED Not Lit 7-9
 - 7.5.3 Connectivity Problems 7-9
 - 7.5.4 ATX Has Rebooted 7-10
 - 7.5.5 ATX Does Not Respond To NMS.....7-10

**CHAPTER 8 ADDING/SWAPPING MODULES AND
MAINTENANCE**

- 8.1 ADDING A MODULE.....8-1
- 8.2 SWAPPING A MODULE.....8-2

8.3 MAINTENANCE.....	8-3
8.3.1 Power Fuse.....	8-3
8.3.2 Fan Filters.....	8-4
8.3.3 Hot Swapping the Power Supply	8-4

APPENDIX A SPECIFICATIONS FOR THE ATX

APPENDIX B PACKET TRANSLATION PROCEDURE

APPENDIX C NULL MODEM CABLE PINOUTS

APPENDIX D GLOSSARY

APPENDIX E BIG ENDIAN TO LITTLE ENDIAN CONVERSION

CHAPTER 1

INTRODUCTION

Welcome to the Cabletron Systems ATX User Guide. This manual explains installation instructions, and provides specifications for the ATX.

1.1 USING THIS MANUAL

This manual is for system administrators responsible for configuring, monitoring, and maintaining the ATX.

You should have a familiarity with internetworking concepts and principles when you install the ATX. A basic understanding of SNMP is helpful. Additionally, if you are using IP routing, you should have an understanding of how to assign addresses. The incorrect use of IP addresses can cause problems on your network as well as across the Internet if you are connected to it. A list of reference material is provided in the section **Related Documentation**.

This manual is the base of the ATX documentation set. Each module that you can use in the ATX also has its own manual. The complete documentation set is described in the section **Related Documentation**.

Much of the configuration of the ATX needs to be done using an SNMP-based network management station, therefore, how you configure is dependent on the station you use. Where applicable, this manual provides instructions for using the ATX's Local Console Manager (LCM) to perform basic configuration. Where it isn't possible to use LCM, general instructions and guidelines applicable to most network management stations are provided.

The contents of each chapter are described below.

- Chapter 1, **Introduction**, provides an overview of the ATX architecture, bridging and routing functions, and describes the Local Console Manager and its command syntax.

- Chapter 2, **Installing and Connecting to the Network**, describes the ATX front panel, how to install the ATX, and how to connect the Local Console Manager.
- Chapter 3, **Configuring**, provides instructions for configuring bridging, and IP, IPX, and AppleTalk Phase II routing using the Local Console Manager. It also provides the MIB variables for configuring multicast storm protection and some common variables you may want to change.
- Chapter 4, **Monitoring and Managing the ATX**, describes how to monitor status and statistics. It also describes how to manage modules and ports using the Local Console Manager.
- Chapter 5, **Filters**, provides instructions for adding, modifying, and deleting filters using the Local Console Manager. It also provides specific examples of how filters can be used.
- Chapter 6, **Traps**, describes the traps the ATX sends to an SNMP manager.
- Chapter 7, **Diagnostics and Troubleshooting**, describes the ATX diagnostics and provides information on troubleshooting common problems.
- Chapter 8, **Adding/Swapping Modules and Maintenance**, provides instructions for adding or swapping a module. It also describes how to change fuses and clean the fan filters.
- Appendix A, **Technical Specifications**, lists ATX specifications.
- Appendix B, **Packet Translation Procedure**, describes the canonical format the ATX uses for translating packets.
- Appendix C, **Null Modem Cable Pinouts**, provides the cable pinouts for a null modem cable.
- Appendix D, **Glossary**, provides a glossary of terms both specific to the ATX and common to the internetworking field.

- Appendix E, **Big Endian to Little Endian Address Conversion**, describes how to convert MAC addresses from big endian (Token Ring native) to little endian (Ethernet) format.

1.2 DOCUMENT CONVENTIONS

The following conventions are used in presenting information in this manual:

LCM commands, prompts, and information displayed by the computer appear in Courier typeface:

```
Current Number of Static Addresses:    5
Current Number of Learned Addresses:  133
Number of Defined Filters:             4
```

Information that you enter appears in Courier bold typeface:

```
ATX >status
```

Information that you need to enter with a command is enclosed in angle brackets < >. For example, you must enter a MAC address to execute the `address matrix <MAC address>` command:

```
ATX >address matrix 00:40:27:04:1a:0f
```

Field value options appear in bold typeface. For example, a filter type can be either **Entry** or **Exit**.

Note: *The Note calls the reader's attention to any item of information that may be of special importance.*

Caution: *A Caution alerts the reader to a specific action which may negatively affect your computer equipment, server communication with your ATX, or may cause data loss.*

Warning: *A warning means you could cause physical harm to yourself. Follow the guidelines in the manual or on the unit itself when handling electrical equipment.*

1.3 RELATED DOCUMENTATION

You may need to refer to the following documentation:

- *ATX MIB Reference Guide* – contains enterprise MIB information.
- *Token Ring Switch Module User Guide* – contains instructions on installing the modules into the ATX and connecting your Token-Ring module to the network.
- *FDDI Dual-Attached Intelligent Module User Guide* – contains instructions on installing the modules into the ATX and connecting your intelligent FDDI module to the network.
- *Fast Ethernet Switch Module User Guide* – contains instructions on installing the modules into the ATX and connecting your Fast Ethernet modules to the network.
- *Ethernet Switch Module User Guide* – contains instructions on installing the modules into the ATX and connecting your Ethernet module to the network.

If you need internetworking reference material, you may find the following books helpful:

- *Interconnections, Bridges and Routers*, Radia Perlman, Addison Wesley © 1992.
- *Internetworking with TCP/IP: Principles, Protocols, and Architecture* (2nd edition), Volumes I and II, Douglas Comer, Prentice Hall © 1991.
- *Inside AppleTalk* (2nd edition), Gursharan S. Sidhu, Richard F. Andrews, Alan B. Oppenheimer, Addison-Wesley © 1990.
- *The Simple Book, An Introduction to Management of TCP/IP-based internets*, Marshall T. Rose, Prentice Hall © 1991.

1.4 GETTING HELP

If you need additional support related to this device, or if you have any questions, comments, or suggestions concerning this manual, contact Cabletron Systems Technical Support:

Phone:	(603) 332-9400 Monday – Friday 8 A.M. – 8 P.M. Eastern Time
CompuServe:	GO CTRON from any ! prompt
Internet mail:	support@ctrone.com
FTP:	ctrone.com (134.141.197.25) Login: anonymous Password: your email address
BBS:	(603) 335-3358
Modem setting:	8N1: 8 data bits, No parity, 1 stop bit

Before calling Cabletron Systems Technical Support, have the following information ready:

- A description of the failure
- A description of any action(s) already taken to resolve the problem (e.g., changing mode switches, rebooting the unit, etc.)
- A description of your network environment (layout, cable type, etc.)
- Network load and frame size at the time of trouble (if known)
- The serial and revision numbers of all modules in the ATX
- Module status (crash codes, if any), firmware version, any verbose display messages; to display messages, use the **display verbose** and **status** commands
- The device history (i.e., have you returned the device before, is this a recurring problem, etc.)

- Any previous Return Material Authorization (RMA) numbers

For additional information about Cabletron Systems products, visit our World Wide Web site: <http://www.cabletron.com/>

1.5 ATX ARCHITECTURE

The ATX is a high-performance, multi-protocol, LANswitch providing multi-technology, multi-layer switching capacity, performance and intelligence, creating a unique platform for LAN to ATM migration.

The ATX has five slots for various interface modules and space for two power supplies. The ATX front panel is shown in Figure 1-1.

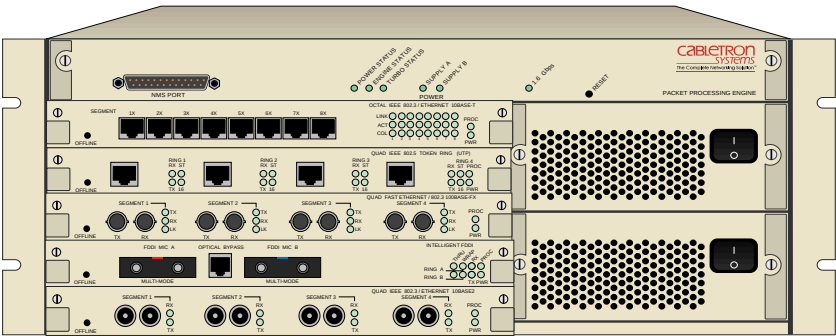


Figure 1-1. The ATX Front Panel

1.6 ATX FEATURES

Cabletron Systems ATX is designed to meet the growing demands for bandwidth across the enterprise-wide network. The ATX integrates the functions of a translation bridge, router, and concentrator/repeater into a single unit. It is designed to support multiple independent networks which are internally bridged and/or routed together with the level of reliability required of mission-critical networks. The internetworking function is performed by a high performance RISC processor-based Packet

Processing Engine.

The ATX offers features which allow you to easily manage and maintain your network, such as:

- Protection against multicast storms.
- Data flow control based on packet filters that you define.
- Compilation of statistics for traffic generated by each user device connected to an ATX segment.
- Ping and Trace Route provide the ATX with the ability to execute (through LCM) ping and trace route commands which show router hops, IP interfaces each packet must traverse and how much time elapsed between transmit and response of a ping command. For additional information on Ping commands, see section 3.13, *Ping Commands*. For additional information on Trace Route, see section 3.14, *Trace Route Commands*.
- Power supplies and input/output modules that can be swapped without disrupting operation of the ATX.
- Configuration and management using the Simple Network Management Protocol (SNMP) with either an in-band or out-of-band connection.

The ATX includes many functions presently available only in bridges or routers. It offers much greater throughput to users, since each module is an independent network and the traffic from a module or network is not repeated to the others as is done in many hubs.

As a bridge, the ATX provides high throughput for each network connected to its ports, translates user-selected packets, and implements the IEEE and IBM Spanning Tree protocol.

As a router, the ATX implements a suite of IP routing protocols, including IP, ARP, Reverse ARP, Proxy ARP, RIP, and IP multicasts. The ATX also implements IPX routing using RIP and SAP. Additionally it implements AppleTalk Phase II routing.

With an innovative, multiple RISC processor architecture, the ATX's Packet Processing Engine is capable of filtering and forwarding at full line speed. Further, the ATX's protocol-independence and high performance allow for transparent, plug-and-play network operation. The ATX offers all the benefits of interconnecting LANs across a backbone with an increase in performance over existing bridges.

1.6.1 Netbios Name Caching

The ATX provides the capability of transforming certain Netbios broadcast frames into non-broadcast frames. The specific frames handled by Netbios Name Caching are those which seek to locate another netbios station. These include Datagrams, Name Query, and Name Recognized frames. For Netbios Name Caching to function, it must be enabled on all ports for which Netbios traffic exists.

When the ATX receives any of these frames and Netbios Name Caching is enabled on the port the frame was received on, the ATX will identify the frame as a special Netbios Name Caching frame. Once identified, a couple of actions takes place. First, the ATX learns the Source Netbios name, the MAC address of the source workstation, which port the station lives on and any applicable RIF information. Second, the ATX determines if the destination Netbios name has been learned. If the Netbios name is learned, then the ATX replaces the broadcast address with the learned unicast address, constructs an appropriate RIF is applicable, and directs the frame to the appropriate port.

The ATX possesses name caching, the ability to reduce the amount of broadcasts of certain Netbios session initialization frames. Name Caching works by using certain frames (Name_Query request and Name_Recognized response) within the Netbios architecture to identify workstation names and their respective hardware MAC address. Once the ATX identifies a workstation and its hardware MAC address, the workstation no longer needs to flood broadcasts to locate a particular destination on the network; the ATX replaces the broadcast address with the learned

unicast address.

Name_Query_Request frames provide the ATX with the name of the source workstation, the MAC address, the port which recieved the frame and any applicable RIF information. The Name_Recognized_Response provides the ATX with information including the name of the workstation, the MAC address of the workstation and any applicable RIF information.

Note: *If cached information on the originating workstation has not timed out, the Name_Recognized will be a directed response instead of an all-stations broadcast. If the workstation name has not timed out from the Netbios Name Cache, the next Name_Query frame destined for either workstation is sent as a directed frame instead of a single route broadcast.*

1.6.2 ATX Local and Remote Port Mirroring

Port mirroring allows the ATX LAN switch to redirect network traffic (excluding MAC layer errors) from one or more ports to any other port, in effect “mirroring” all network traffic to a selected port. This feature allows customers who have existing investments in external analyzers, external RMON probes, or devices like Network General’s Distributed Sniffer System to continue to receive expert analysis and packet decode functions in a switched environment - simply use the port mirroring function to mirror switched traffic to the designated “diagnostic” port to which the analyzer is attached.

The ATX LAN Switch supports local and remote port mirroring. Local port mirroring is when the diagnostic port is on the same ATX as the mirrored ports. Remote port mirroring is when the diagnostic port is on a different or remote ATX from the mirrored ports. The mirrored ports have to be either local or remote to the diagnostic port, not both. In the case of remote mirroring, the traffic from the mirrored ports is encapsulated into an IP packet and sent to the IP destination defined (the diagnostic port). See section 3.11 *Local and Remote Port Mirroring Commands* for additional information on Port Mirroring commands.

1.6.3 IPX with Token Ring Source Routing

Token ring networks often interconnect with source routing (SR) bridges. Although the source routing is a MAC layer feature, all packets must provide the correct source route information to the bridges in order to traverse the networks. To successfully and efficiently route network traffic in such environments, routers need to have the capability to explore and select routes, cache and age route information, and construct network packets with the proper route information. Support of IPX over source routing (IPX SR) enables the ATX LAN switch to achieve this capability and route IPX packets through SR bridges. See section 3.11 *IPX Routing Over Source Route Commands* for additional information on source routing commands.

1.6.4 Event Logging on the ATX

Event Logging is an ATX troubleshooting tool. It records selected classes of networking events then analyzes the log of events recorded to assist in diagnosing problems on the network. ATX Event Logging includes the following features:

- Separate enabling flags for each event or class of events. The enabling flags are symbolic and are thus easily used in troubleshooting the network.
- Continuous monitoring of events is supported.
- Logging entries are easy to add and delete from the source code.
- The framework is integrated with SNMP and easily fits into the anticipated fault/ alarm restructuring.

See section 3.14 *Event Logging Commands* for additional information on Event Logging.

1.6.5 ATX LAN Switch Workgroups

Virtual workgroups allow you the flexibility to control broadcasts in the network. By reducing broadcasts throughout the network, it

preserves network bandwidth for important user data and frees up valuable end station processing. By defining virtual workgroups, broadcasts will only be seen by other end stations within the same virtual workgroup. With the functionality to define workgroups by port grouping, IP network address and/or IPX network number, a station can be part of multiple workgroups based on their location and protocol.

Each workgroup can be defined by port, IP network address and/or IPX network number. A total of 100 virtual workgroups can be defined on each ATX LAN Switch. The ATX LAN Switch can route between IP workgroups but all other workgroups will need an external router (See Workgroup to Workgroup Communication). For additional information, see section 3.8 *Virtual Workgroup LCM Commands*.

1.6.6 ATX Packet Processing Engine

The ATX architecture, diagrammed in Figure 1-2, is based on dual 29030 RISC processors on the Packet Processing Engine version 3(PPE-3). In addition, it includes the following:

- At least one RISC processor per i/o module
- Backplane providing 1.6 Gbps capacity, with a load balancing architecture for maximum accessibility for I/O modules
- A 2mb shared RAM architecture, which is optimized using adaptive buffer allocation. Adaptive Buffer Allocation (ABA) is an algorithm providing a sophisticated distribution of packet buffering to meet varied utilization demands per port.

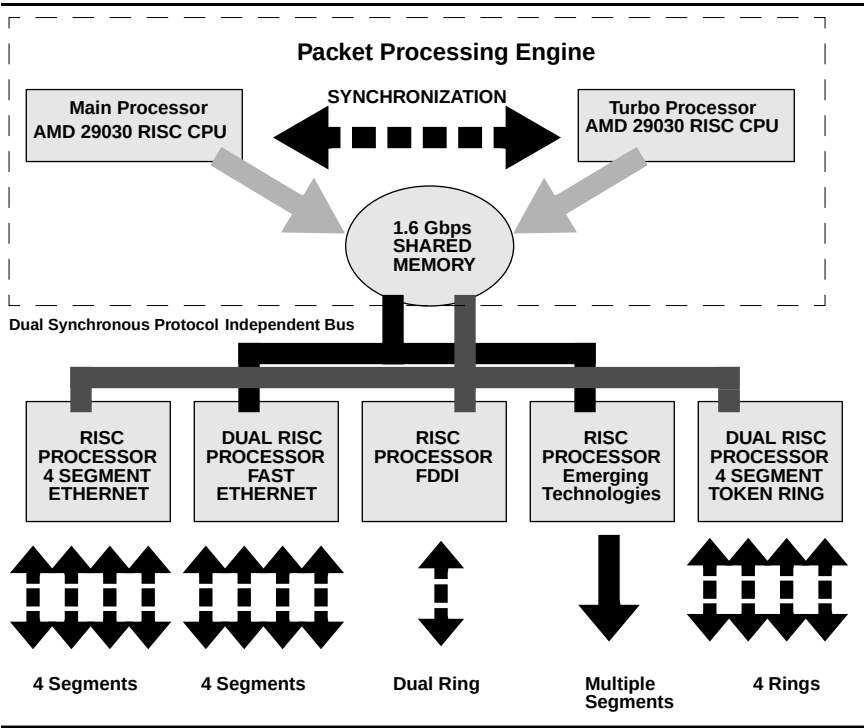


Figure 1-2. ATX Architecture

1.6.7 Input/output Modules

The ATX has four types of modules available. The modules slide into the face of the ATX. The module installation procedures are in Chapter 8.

The ATX supports the following:

- 3E02-04, 3E05-04, 3E07-04, 3E08-04, and 3E02-08-ATX - Multi-segment Ethernet modules that come in five models-four UTP 10BASE-T connections, four AUI connections, four BNC 10BASE-2 connections, four fiberoptic 10BASE-FL connections, and eight UTP 10BASE-T connections respectively.

- 3T02-04, 3T05-04 and 3T01-04 - Four ring Token Ring modules accepting data frames from and sending data frames to four Token Ring networks. The 3T02 and 3T01 modules support UTP and STP cable types respectively, while the 3T05 supports either UTP or STP.
- 3F00-01 and 3F55-01 - DAS (dual-attached station) FDDI modules. These modules transfer packets from and to a FDDI network. The front panel accepts media interface connectors (MICs) for multi-mode fiber (MMF) such as the 3F00-01, or single mode fiber (SMF) such as the 3F55-01. Both modules support an external optical bypass switch (OBS). Each has a built-in DMA controller, but not a general purpose processor, so the station management functions are performed by the PPE.
- 3H02-04 and 3H08-04 - Four port 100 Mbps Fast Ethernet Switch modules. These modules support UTP via RJ71 connectors and fiberoptic via ST connectors respectively.

Modules are described in greater detail in the documentation that accompanies each module.

1.6.8 Power Supply

The ATX comes with one self-ranging power supply. An optional redundant power supply is also available that automatically takes over when the primary power supply fails. Each power supply has its own power entry module and fuse assembly to allow the use of separate power sources. When both supplies are used the load is balanced between the power supplies.

1.7 BRIDGING FUNCTIONS

The basic bridging function of an ATX is to transparently forward data packets to the network segments (LANs) it interconnects. Incoming packets are stored momentarily while the ATX checks their destination addresses against the ATX's address table. If a packet's destination address is not on the same network segment

as the originating packet, the ATX immediately forwards the packet to the segment associated with the destination address. Local traffic, data packets whose source and destination address is on the same segment, is automatically discarded.

The ATX forwards data packets to network segments based on the IEEE 802.1D spanning tree algorithm, which converts multiple LANs into a “spanning tree” of networks. This standard defines a logical (not physical) network configuration consisting of one extended LAN without active duplicate paths between ATXs. The ATX and other spanning tree compliant bridges in the network dynamically configure the network topology into a single spanning tree by exchanging bridge protocol data units (BPDUs).

In a parallel configuration of bridges packets are forwarded to LANs by only one ATX (or other spanning tree compliant bridge). When there are multiple ATXs between two LANs, only one of the ATXs forwards any individual packet. The spanning tree algorithm determines which ATX should forward each packet.

Packets originating from one device and destined for a remote device are forwarded in the same order in which they are received.

Each port of the ATX can be configured for transparent (802.1d) bridging, IBM source routing bridging, or source routing transparent bridging (802.5M). Depending on network topology, it may be desirable to include a mix of these methods within a single ATX.

The choice of bridging methods is determined both by end station requirements and by other internetworking equipment.

Source routing end stations may use any of the ATX three bridging methods. Transparent end stations must use either transparent or SRT bridging. When in doubt, transparent bridging is the easiest to configure and use.

If redundant links are employed along with IBM source routing bridges, then the attached ATX port should be configured for source routing. This will enable the mesh of bridges to derive a spanning tree suitable for spanning tree explorer frames and for

multicast packets.

If source routing is desired, and either Ethernet or FDDI is to be used as a backbone between Token Rings, then the Ethernet or FDDI port should be configured for SRT bridging. (SRT over Ethernet is not a standard, but is available for use between multiple ATX chassis in backbone applications. In this case, the "Ethernet" may actually be a microwave or satellite link with an Ethernet-like interface.)

A common mixture of bridging modes may occur when Ethernet segments and Token Ring segments do not exchange data but share an FDDI backbone. In this case, the Ethernets may be configured for transparent bridging, the Token Rings for source routing, and the FDDI backbone for SRT. (Don't infer from this example that SRT is the sum of transparent and source routing bridging; it is a distinct third method).

The bridging method is dependent on the configuration of the bridge entry and exit ports, and the value of the Routing Indicator (RII) bit in the received frame. The following chart summarizes the interaction between the bridging method.

		Exit Port Configuration		
Entry Port Config.	RII	SRT (Source Routing Transparent)	SR (Source Routing)	TST (Transparent Spanning Tree)
SRT	0	spanning tree	block	spanning tree
	1	source route	source route	spanning tree ^a
SR	0	block	block	block
	1	source route	block	block
TST	0	spanning tree	block	spanning tree
	1	spanning tree ^a	block	spanning tree ^a

a. source address is not learned

1.7.1 Transparent Bridging

Transparent or spanning tree bridging requires no initial programming. After being installed on the network, bridges “learn” and remember the location of the attached devices by reading the source addresses of incoming packets. Then they place the source address and port information in a lookup table.

When a packet comes into a port, the bridge reads the destination address and attempts to find the location of the destination node using its lookup table. If the address is in the table, the bridge simply re-transmits the packet out of the appropriate port. If the address is not found in the table the bridge re-transmits the packet out of all the ports except the source port.

Transparent or spanning tree bridges also usually provide some packet filtering capabilities. On some networks it is desirable to prevent certain stations from accessing other segments. The ATX uses this bridging method.

1.7.2 Source Route Translational Bridging

Source Route Translational Bridging (SRTB) allows the ATX to strip and cache routing information for source route frames. Routing information (RIF) is used in source route networks to indicate the path a frame has taken through the network. This feature will enable the ATX to switch between source route only networks like Token Ring and transparent networks like Ethernet and FDDI. RIF is not supported on Ethernet networks and is seldom used on FDDI networks. In order to merge source routed Token Ring networks with transparent Ethernet and FDDI networks the ATX must strip the RIF when communicating to Ethernet or FDDI and insert the RIF when communicating back to Token Ring. SRTB on the ATX contains the following features:

- A redundant/load sharing source route network is NOT supported when SRTB (RIF caching) is enabled. A redundant/load sharing source route network could have multiple paths to the transparent network and cause the

learning database to learn addresses on the incorrect ports. This could result in frames not getting forwarded and loss of communication.

- SRTB is a global parameter and is enabled only on Token Ring ports with SRT bridging mode.
- The RIF database supports 8,192 entries.
- SRTB can be enabled based on IP, IPX and other protocols (SNA, NetBIOS, etc.)
- All existing protocol translations (IP, IPX, SNA, NetBIOS and AppleTalk) are supported when SRTB is enabled.
- The RIF caching aging timer is the same as the Spanning Tree timer, and is configurable. The default value is 300 seconds.
- The RIF cache entry is relearned based on a separate timer that is set to one half the Spanning Tree timer.

1.7.3 Source Routing Bridging

Source routing bridging (SR) is an alternative to transparent or spanning tree bridging, and is widely used in Token Ring networks. The ATX supports source routing bridging on Token Ring LANs, and an enhancement to source routing called SRT on all LANs.

With source routing bridging, all networked devices participate in the source routing protocol. Each packet that crosses a bridge specifies the originator's LAN segment, the particular bridge, and the destination LAN segment. It may also specify intermediate LAN segments and bridges.

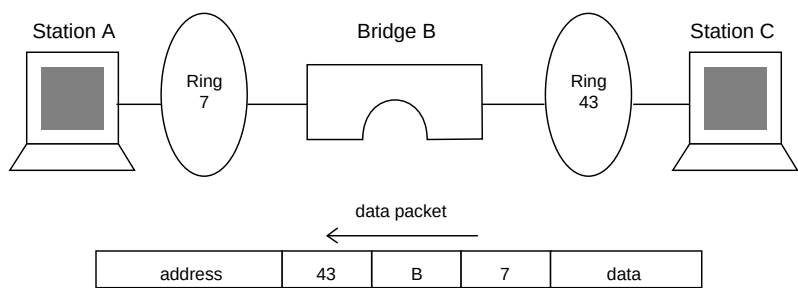


Figure 1-3. Source Routing Example

In the example in Figure 1-3, a data packet traveling from station C on LAN 43 through bridge B to station A on LAN 7 must specify the full route it is to take. The source station is responsible for specifying the route, hence the term “source routing.”

Bridges in a source routing network must be configured with the LAN numbers (normally 1 to 4095) to which it is connected and a bridge number (normally 1 to 15). The network administrator chooses the numbers; the LAN numbers must be unique in the source routed network and the bridge numbers must be unique between each pair of LANs.

Source routing workstations need not be configured with route information; instead they discover the best route to a destination through the use of explorer frames. In the Figure 1-3 example, station C might first transmit an empty explorer frame. Bridge B would add 43-B-7 as its portion of the route, and then transmit the explorer on all other LANs. When the packet reaches station A, it can reverse the route to send a reply back to C. When C receives the reply, both stations have all of the routing information needed to converse, with no further explorer frames needed.

Part of the original intent of source routing bridging was to enable LANs to be richly connected by low-performance, low-cost bridges. As shown in Figure 1-4, source routing allows an end station to choose a less-congested path through a chain of bridges,

where each bridge is likely to become congested.

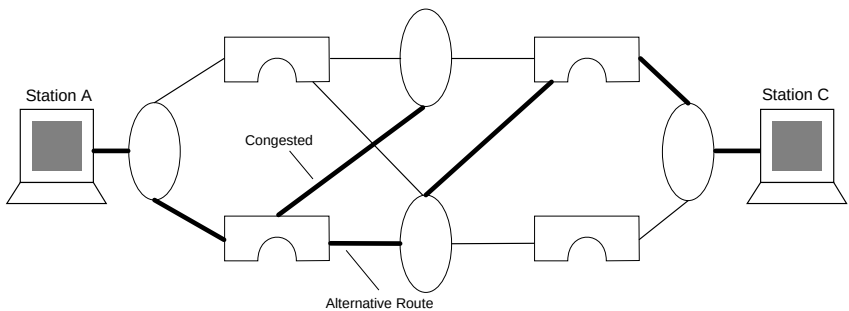


Figure 1-4. Data Path Using Source Routing Bridging

In contrast to spanning tree bridging, all bridges and all links are active with source routing bridging; the least-congested path is chosen at discovery time. With products like the ATX, such congestion avoidance is rarely necessary, since the bridge is capable of handling nearly any traffic load without experiencing congestion.

1.7.4 Source Routing Transparent Bridging

Source Routing Transparent (SRT) bridging is a method that merges IBM-style source routing with transparent spanning tree bridging. If a route is present in a packet, then the bridge uses it; otherwise the bridge applies transparent learning rules. It represents an attempt by the IEEE standards committee to standardize source routing and correct some shortcomings in source routing (notably multicast transmission). IEEE has defined SRT bridging for Token Rings, and ANSI has incorporated it into the FDDI standards. The ATX supports SRT bridging on these, as well as on Ethernet (for Ethernet, there is no such standard; the ATX provides this as a proprietary backbone service).

1.7.5 Translation

The ATX is a translating bridge; meaning it translates packets across unlike protocols. For example, if an Ethernet (802.3) data packet is to be forwarded to an FDDI segment, the ATX translates the packet to FDDI packet format. Conversely, the ATX translates FDDI packets to be forwarded to an Ethernet segment to Ethernet (802.3) packet format. This means the ATX can transparently exchange data packets between FDDI and Ethernet LANs.

A translation parameter was added to the ATX Token Ring module to complete the StripRIF protocol set, StripRIF ALL. The current set of protocols that StripRIF is used for are: IPX, ARP, NetBIOS and SNA. This feature allows the ATX to handle protocols which need the RIF stripped before being transmitted out other ATX LAN switch ports (for example, protocols such as BootP and RIP). For additional information on StripRIF commands, see the section *1.10 Basic LCM Commands*.

For example, in Figure 1-5 workstation 100 on Ethernet LAN 1 is able to access server F1, which is attached directly to the FDDI ring, or server E1 on Ethernet LAN 2. To reach either server, the Ethernet packet from workstation 100 is translated by ATX A to a FDDI format. To reach server E1, the packet is translated by ATX B back into Ethernet format.

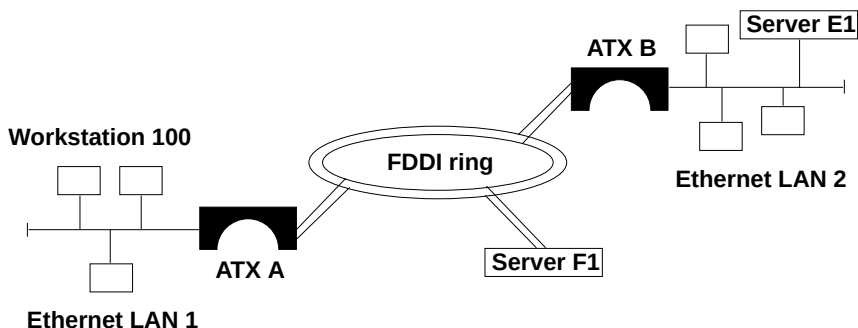


Figure 1-5. Network Where Translation Must Occur

The ATX uses a standardized internal format called canonical format, for packet translation. (Refer to Appendix B, **Packet Translation Procedure** for an explanation of the packet translation procedure.) The ATX converts all incoming packets into its internal format and then converts each packet from its internal format to either FDDI, Ethernet, or Token Ring format, depending on the packet's destination.

The ATX can interoperate with other vendors' translating bridges. Translation allows end-nodes to reach destinations on the FDDI ring as well as destinations attached to other vendors' translating bridges and routers. In Figure 1-5 for example, ATX A or ATX B could be bridge products from other vendors.

1.8 ROUTING FUNCTIONS

The ATX can route packets that use the IP and IPX, and Appletalk protocols. A brief overview of these follows. For more in-depth discussions, refer to the books listed in the section, **Related Documentation**.

Note: *When the ATX is not configured for routing, it's necessary to establish a default gateway so that management can take place using a SNMP agent. To establish a default gateway connection, apply the route add command through LCM using the following format: **route add IPaddr Gwaddr PORT# [hops] [IPmask]**. For example: **route add 0.0.0.0 176.16.107.19 3** This command establishes port 3 as the default gateway to the router at 176.16.107.19. Any port (2 to 41) can be the default gateway. The IPaddr 0.0.0.0 signals that this is the default gateway specification. Other addresses can be used to explicitly and statically route some IP traffic while remaining in bridging (rather than routing) mode. SNMP management stations are now able to poll the ATX locally and remotely, but this does not permit the ATX to send SNMP traps to multiple SNMP management stations. To identify a specific SNMP management station where traps are sent, change the [configNMSAddress] MIB located in the MIB tree at:*

private/enterprise/sigma/ecs1/admin/config. The default setting for this MIB is 0. Query the MIB and change this value to the address of the SNMP management station, then SET. If the [configNMSAddress] MIB is not changed, traps are sent to the last SNMP manager which polled the device.

1.8.1 IP Routing

IP routing allows end-nodes to send packets to end-nodes elsewhere on the network using the IP protocol suite. The ATX builds an IP routing table that stores destination addresses, the address of the gateway through which that destination can be reached, and the number of hops it takes to get there. (The number of hops is the number of other routers or gateways a packet must go through to get to the gateway.) The routing table allows the ATX to know how to route a packet to reach its destination address.

The routing table is a dynamic table, meaning that entries are continually being added and timed-out based on information the ATX is receiving.

Routing Information Protocol (RIP)

RIP is one of the protocols that allows the ATX to build an accurate, current routing table. Routers, including the ATX, send out broadcasts every 30 seconds advertising the networks they know about, the routes to those networks, and the number of hops to get there. In this way the ATX is constantly up-to-date on the state of its neighboring networks.

Address Resolution Protocol (ARP)

ARP provides a method for mapping IP addresses to physical hardware addresses. When a device wants to communicate with another device whose hardware address it doesn't know, it sends out an ARP request. An ARP request contains the IP and hardware addresses of the source, and the IP address of the potential

destination device. If the device is on the network, it will respond with its hardware address.

Reverse Address Resolution Protocol (RARP)

If the ATX is not configured with an IP address, it uses reverse ARP (RARP), to send out broadcasts of its physical hardware address to find its IP address.

Proxy ARP

Proxy ARP provides a mechanism whereby the ATX can respond to an ARP request on behalf of a device that is located on a network behind it. This is particularly helpful if you are using IP subnetting. The ATX could respond to a request on behalf of devices that it knows about, in effect acting as a proxy agent for that device.

BOOTP

The BOOTstrap Protocol (BOOTP) uses IP to deliver a packet including an IP address, the address of a router and the server address. Enabling the BOOTP relay option is useful in environments where you have a diskless client and its server is on a network on the other side of the ATX. When the client boots up, it sends out a broadcast requesting the software it needs to download. If bootp is not enabled, the ATX won't forward the broadcast to the network where the server is located. This may also be used to relay DHCP frames.

IPM

Enable IP multicasting. IP multicasting is the transmission of IP packets to a host group. A host group is a set of hosts identified by a single IP address.

1.8.2 Multiple IP Networks Per Port

The ATX's routing software allows you to configure a single IP network to span multiple physical network segments (ATX ports). This enables you to configure multiple physical networks as one logical network.

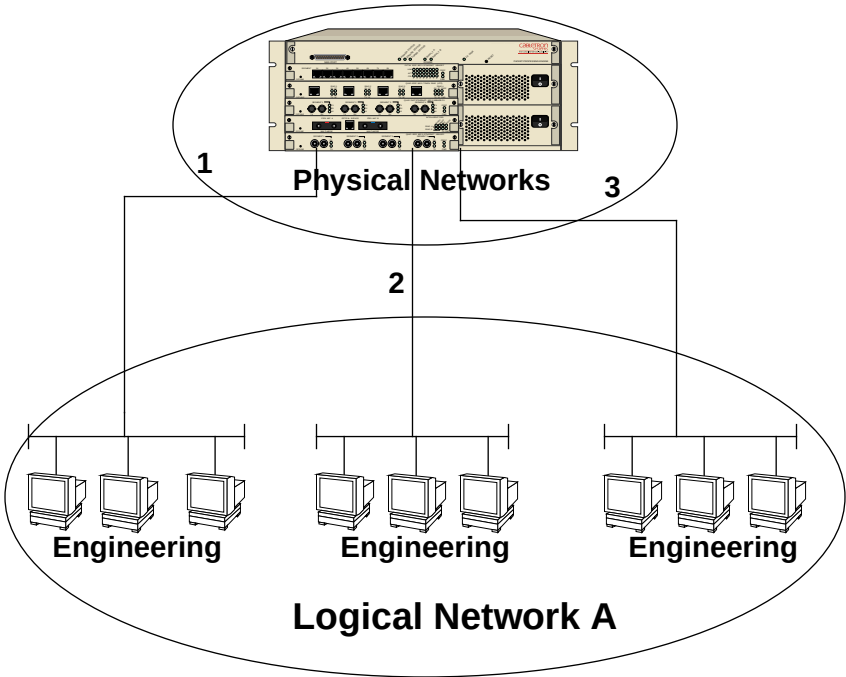


Figure 1-6. One Logical Network On Multiple Physical Networks

The ATX also allows you to assign multiple IP network addresses to one physical network segment (ATX port). This feature allows you to configure two or more logical networks on the same physical network segment. If the hosts on a physical network segment exceed the current logical network's capacity, you can easily add another logical network to the physical network segment.

In addition, by overlapping logical networks, a user who moves to another physical network segment can remain on the same logical network and retain their net/host IP address, even if he or she is sharing the new physical network segment with other logical networks. This is known as address mobility and is a useful Virtual LAN component that can ease adds, moves, and changes and the definition of broadcast domains.

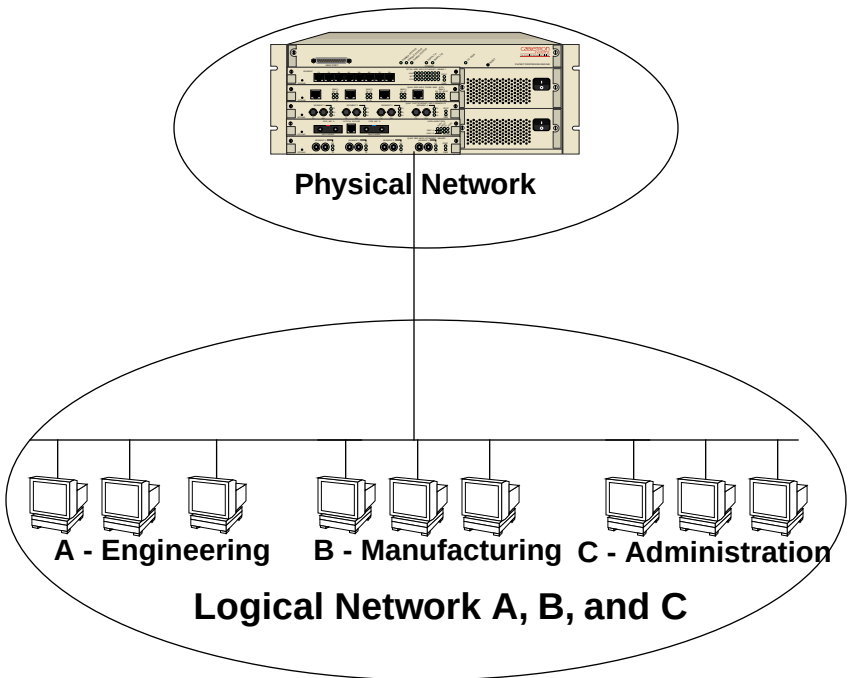


Figure 1-7. Multiple Logical Networks On One Physical Network

When assigning multiple IP network addresses to an ATX port, the port must be configured for routing. In addition, the logical networks connected to an ATX must see the ATX as a gateway(router). The way a host accomplishes this is dependent

upon the operating system or TCP/IP being used. The host becomes aware of a gateway in one of three ways:

- The host is manually configured with a default gateway address.
- The host is listening to Routing Information Protocol (RIP) broadcasts.
- The host is participating in the router discovery protocol (ICMP).

When using LCM each ATX port can be configured for zero or more IP addresses, with associated subnet masks. Each IP address defines an IP subnetwork. Each IP subnetwork is a distinct entity with respect to protocols, such as RIP (Routing Information Protocol), and is treated as a separate interface. Specifically:

- RIP advertisements are transmitted to each IP subnetwork broadcast address of the IP addresses associated with a ATX port. RIP advertisements include route descriptions of the other IP subnetworks assigned to that ATX port. For example, if a ATX port has three IP addresses assigned to it, three RIP advertisements are transmitted each interval, and each RIP advertisement publicizes the other two IP subnetworks.
- Router discovery ICMP packets are transmitted to either the host's IP multicast address, or to the local broadcast address, regardless of how many IP addresses are assigned to an ATX port. All IP subnetworks assigned to an ATX port are advertised in each router discovery ICMP packet.

1.8.3 IP Multicast Routing

The Internet Protocol (IP) is recognized as the base technology for multimedia applications. The implementation of IP multicast routing complies with the DVMRP standard.

In general, IP multicasting is the transmission of IP packets to a *host group*. A host group is a set of hosts identified by Class D IP

addressing (i.e., those IP addresses with 1110 as their high-order four bits). Using Internet standard dotted decimal notation, host group IP addresses range from 224.0.0.0 to 239.255.255.255. The IP address 224.0.0.1 is assigned to the permanent group of all IP hosts.

Members of a host group can:

- Join and leave the host group at any time
- Be included as a member in multiple host groups.

A host group can be permanent or temporary. A permanent host group maintains a single IP address regardless of how many members it has. A temporary host group is one that must have at least one member, a permanent host group can exist with zero members. Currently the ATX supports only temporary host groups.

***Note:** There are no restrictions on the location or number of members assigned to a host group.*

IP multicasting provides the following benefits:

- When the same information must be sent to more than one destination, IP multicasting reduces both network overhead and the time it takes for all destinations to receive the information.
- When information must be sent to one or more hosts whose address is either unknown or changeable, IP multicasting can reduce the need for complicated configuration files because permanent host groups maintain a single IP address.

Group membership reports are not forwarded across the network. Instead, routers learn of the existence of other routers on the network, and forward all IP multicast packets to the downstream neighboring router.

When a route receives an IP multicast packet, it verifies the route's origin, and then forwards the IP multicast packet only if there is a downstream neighboring router and/or there is a host group member on the outgoing port.

For example, in Figure 1-8, LANs B, C, and D are bridged to backbone LAN A. A packet originating from LAN C destined to the host group member on LAN B will traverse LANs C and A but not LAN D. Similarly, an IP multicast packet destined to the group member on LAN B that originated on that LAN will not be forwarded to the other LANs.

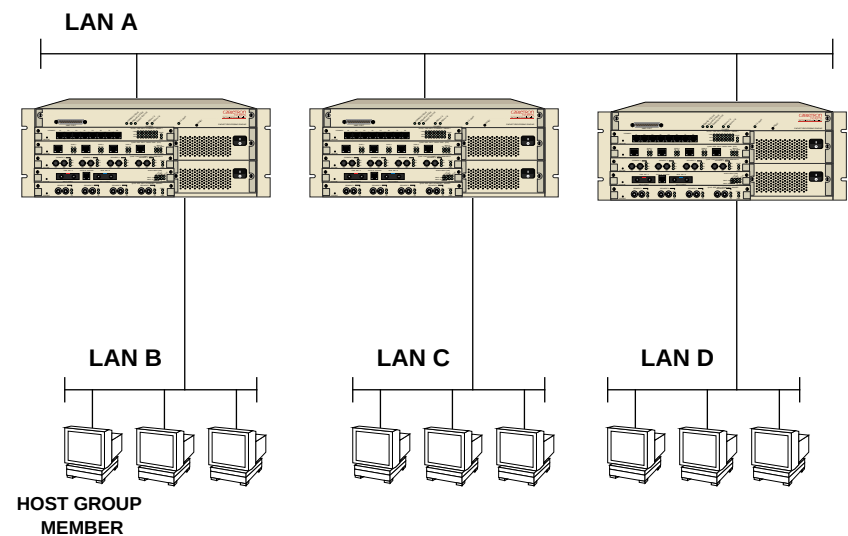


Figure 1-8. Bridged LAN With One Host Group Member

In Figure 1-9, a member on LAN C joins the host group. In this case, IP multicast packets to all host group members on LANs B and C will again traverse LANs A, B, and C, but not LAN D.

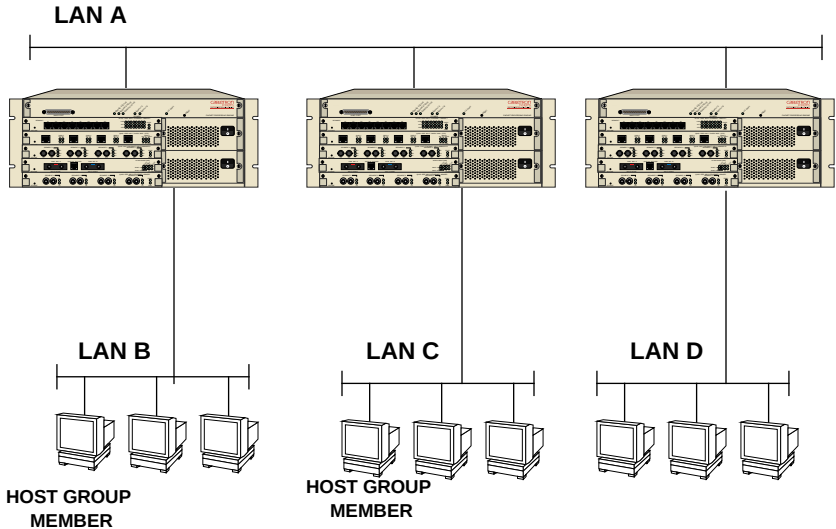


Figure 1-9. Bridged LAN With Two Host Group Members

1.8.4 IP Routing Over Source Routing

Token Ring networks are often connected by source-routing bridges. End-stations that communicate across a source-routing bridging domain must be able to build routes from themselves to their destinations, specifying the Token Rings to be traversed.

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

← Source-routing operates at Layer 2

Figure 1-10. OSI Reference Model

The architecture behind source-routing bridges is that a packet header containing a route is inserted by the source end-station. For the source end-station to discover a route to a destination end-station, it must learn of a route by transmitting a special type of packet called an *explorer* packet. The explorer packet is duplicated by source-routing bridges as it discovers possible route choices.

A copy of the explorer packet is sent over every possible route. When a source end-station discovers a route to a destination end-station, it stores the route so that it can be used for subsequent packets to the same destination end-station. Generally routes are stored for approximately 15 minutes, or three times the Spanning Tree age. However, the time can be shorter if the Spanning Tree topology is changing.

In simplest terms, the data link layer header of a packet on a Local Area Network (LAN) looks like the following.

destination	source	data
--------------------	---------------	-------------

Figure 1-11. Data Link Layer Packet Header

To distinguish between packets whose data link headers include routing information and those that do not involves setting the Routing Information Indicator (RI). The Routing Information Field (RIF) contains the additional source-routing information.

The RI happens to be the multicast bit in the source field. Therefore, a packet with a multicast bit set to 0 is not treated as a source-routing packet. However, if the multicast bit is set, the information following the usual data link layer header is assumed to be a source-routing header.

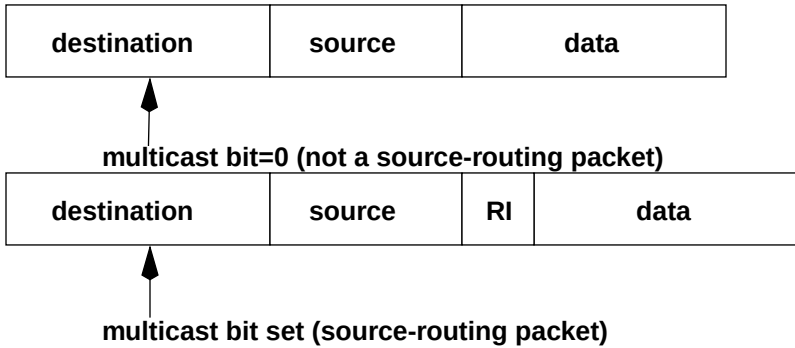


Figure 1-12. Packet Headers With And Without Source-routing Bit Set

In TCP/IP hosts, an explorer packet exchange is normally accomplished as part of the Address Resolution Protocol (ARP). ARP is used to dynamically map IP addresses to MAC addresses. The resulting source route is kept as part of the ARP cache.

The IP routing over source-routing feature allows the ATX to:

- Recognize Type-6 (IEEE 802) ARP packets, as well as Type-1 (Ethernet) ARPs.
- Recognize ARPs received with a null Routing Information Field (RIF) even if the interface is not configured for source-routing.
- Cache the RIF on received ARP packets.
- Transmit ARP requests as source-routing explorer packets.
- Strip RIFs from received IP packets before processing by the IP Router or the IP host.
- Attach the cached RIF to outbound IP packets when the RIF exists as part of the ARP cache.
- Use the largest frame limit returned in the RIF as the Maximum Transmission Unit (MTU) for the outbound packet.

- Transmit IP multicast packets as single route explorer packets.
- Transmit subnet-specific broadcasts as single route explorer packets.

1.8.5 Configuring IP Routing Over Source Routing

The IP routing over source-routing feature is integrated with the multiple IP networks per ATX port feature. Configuration is specified for each binding of an IP subnet.

- No source-routing - ARP requests are sent as transparent explorer packet. This is normal for non-Token Ring LANs.
- Source-routing - ARP requests are transmitted as source-routing explorer packets. This is the default for most TCP/IP implementations on a Token Ring LAN.
- Both transparent and source-routing - Two ARP requests are transmitted; one transparent and one explorer. This provides the best connectivity where it is not known whether the intended destination is capable of source-routing.
- Default - If the port is Token Ring, both types of ARP requests are transmitted. For other LAN types, ARP requests are sent as transparent frames.

Configuration is provided by the `sippkt` portion of the Cabletron proprietary MIB. The `sipNetToMedia` Table, within the Cabletron proprietary MIB, allows you to manage the source route as part of the ARP cache. See the *ATX MIB Reference Guide* for information on `sippkt` and the `sipNetToMedia` Table.

1.8.6 IPX Routing

The ATX can route Internetwork Packet Exchange (IPX) packets. IPX is Novell's protocol that allows users in a Netware internetwork to communicate. The ATX identifies IPX packets and routes them appropriately.

Routing Information Protocol (RIP)

RIP is one of the protocols that allows the ATX to build an accurate, current routing table. Routers, including the ATX, send out broadcasts every 60 seconds advertising the networks they know about, the routes to those networks, and the number of hops to get to there. In this way the ATX is constantly up-to-date on the state of its neighboring networks.

Service Advertising Protocol (SAP)

SAP provides a method for IPX servers such as file servers to advertise the services they provide. It functions much the same as RIP, but it is the servers which send out broadcasts advertising the services they provide. IPX routers gather the information, maintain a database of services they know about, and broadcast that information to other routers. Clients can then find the servers that provide the services they need.

IPX Routing Over Source Route

Token ring networks often interconnect with source routing (SR) bridges. Although the source routing is a MAC layer feature, all packets must provide the correct source route information to the bridges in order to traverse the networks. To successfully and efficiently route network traffic in such environments, routers need to have the capability to explore and select routes, cache and age route information, and construct network packets with the proper route information. Support of IPX over source routing (IPX SR) enables the ATX LAN switch to achieve this capability and route IPX packets through the SR bridges.

***Note:** This feature is valid only for Token Ring and FDDI ports.*

1.8.7 Appletalk Routing

AppleTalk routing allows end-nodes to send packets to and receive packets from other end-nodes through the use of AppleTalk Phase 2 protocol. The ATX stores a table of routing information it learns through Routing Table Maintenance Protocol (RTMP) packets sent out by other routers. The ATX also sends out RTMP packets to let other routers know of the routes it has learned. By storing the RTMP packets, the ATX knows where to forward packets it receives.

AppleTalk addressing

An AppleTalk address consists of 16 bits of network number, 8 bits of host number, and a zone name.

In AppleTalk routing, a logical network is defined by a contiguous set of network numbers. Routes are therefore generated for a network range instead of to a single network number. Macintoshes know both their address (network number/host number pair) and their network range. A router is not needed to send a packet to a different network number in the same range.

AppleTalk zones

The concept of AppleTalk zones allows Macintoshes to be grouped together logically, independent of network address. Each AppleTalk device such as a Macintosh or a printer belongs only to one zone. There may be more than one zone on a network and a zone may be available on many networks.

In order to establish a session with a network device such as a file server or a printer, a user selects a zone name from the list under their network connection in the Network dialog box in the Control Panel. They would then go to the Chooser on their Macintosh desktop and select a device from the list available. For example, you might select the zone *Engineering* and the device *Laser Printer*. Underlying protocols then map this Network Visible Entity name

to an address. In the ATX implementation, the maximum number of zones that a router may be configured is 22 ports. Each configured zone may be available on any subset of ports.

How a Macintosh learns its address

A Macintosh learns its network address automatically; you don't have to assign addresses. This process is called address acquisition and is performed every time a Macintosh enables its datalink, either automatically at start-up or using the network control panel. This process involves determining a network range from a router (or using network 0 if no router is present) and then choosing a host number on that range.

If the Macintosh had chosen a host number the last time it was rebooted, it tries to use that number again. If it never had a number assigned, it picks any unused number. To determine if a number is available, the Macintosh sends out AARP probes. If a device responds to the probe, a different number is tried until the Macintosh finds an unassigned number.

Once the Macintosh has its address, it sends a request to a router to determine if a previously used zone name is valid. The router may either respond affirmatively or provide the Macintosh with a default zone to use. The Macintosh may later change its zone residence from the network control panel by asking the router for a list of available zones. If no router is present, no zone name is assigned.

How a router learns its address

Routers must also go through the address acquisition process, but in a slightly different manner. The process begins each time a link becomes active. The router first chooses an address in the start-up network range (ff00-fffe) so that it has an address that other routers may respond to before it learns its real network range. The ATX probes to find its network range; it picks a network range and sends out probes to see if it can use that range. Once the router

receives a response, it knows its network range and then performs additional AARP probes to choose a host number. The router then sends RTMP requests to begin building its routing table. Next the router asks other routers for a list of zones so it can create a zone list. Although a router maintains a list of zones, it does not reside in a zone the way an end-node does.

Note: *An AppleTalk internet router cannot have two ports connected to the same network.*

Seed Routers

The first router up on the network must choose its network and zone information. This is called the *seed* router. Any router that is configured with network and zone information is capable of acting as the seed router. When a router discovers, via lack of RTMP response, that no other routers are active on the network, the router uses its configured information to seed the network.

Seeding is an event not a state. Once the first router seeds the network, all other routers respond to all requests whether they were originally configured as the seed router or not. This allows the network to remain operational even if the original seed router goes down. A seed router holds no special status once the network is seeded.

A router must have BOTH a valid network range and zone list configured for a port to enable it to perform seeding. A network range of 0-0 is the default and is considered to be unconfigured. A router with a network range of 0-0 cannot be the seed router.

Also note that at least one, but potentially any number of routers on a network may be configured to be capable of seeding. All routers on a given network that are configured to seed should be configured identically. If this is not the case, an error is flagged and the offending router uses the already seeded information. In order to change zones for a network, all routers on the network must be rebooted in order to re-acquire their addresses.

A router that learns its network address from a seed router shows a status of *garnered*; meaning you did not configure it.

1.9 TRUNKING

If your network configuration requires you to connect two or more ATXs together, but the applications you are running over the network require more than a single LAN connection between ATXs, you can use the built-in trunking feature to increase bandwidth up to 8 times the single LAN connection bandwidth (128 Mbps for Token Ring), without installing additional hardware on your network.

Trunking is a proprietary extension to the 802.1D Spanning Tree algorithm. It enables you to use multiple LAN segments to connect ATXs together, while maintaining first-in, first-out ordering of packets. In addition, if any of the LAN segments configured for trunking become inoperable, those LAN segments are automatically bypassed.

Trunking can be used between devices which support trunking. Currently, it is possible to connect Fast Network 10s to ATXs via Ethernet connections, ATX to ATX via Ethernet, Token Ring, or FDDI connections, or Fast Network 10s to Fast Network 10s.

Figure 1-13 below shows two ATXs connected by four 10BASE-T crossover cables. You can connect up to eight ports for sharing the traffic load. Any additional connected ports beyond the eight ports will become standby ports. A standby port is automatically included in a trunk group when a port currently in the trunk group becomes disabled. The connections must be point-to-point. That is, there cannot be any other devices on the trunked LAN segments.

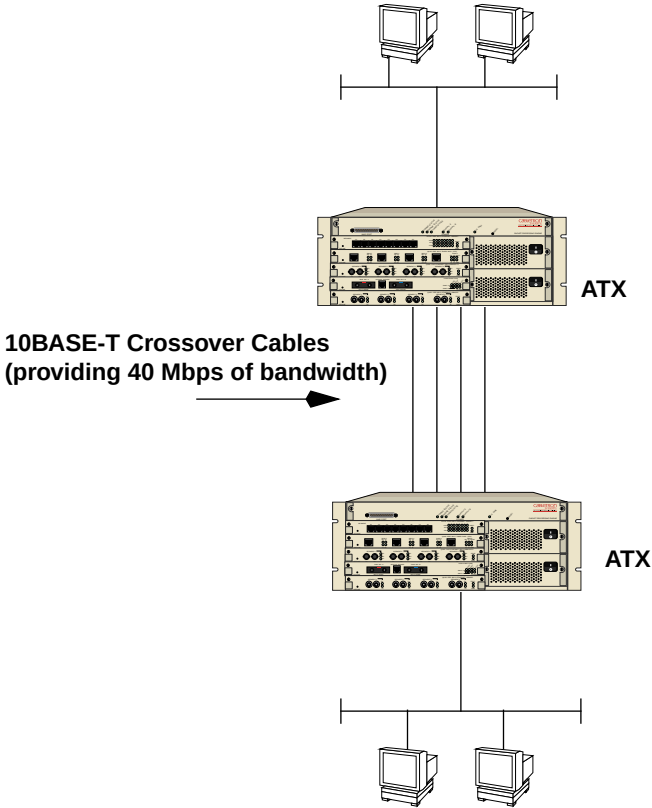


Figure 1-13. Trunk Connections

Trunk Groups

Each set of connections between ATXs is called a trunk group. You can configure several trunk groups to interconnect your ATXs. Each ATX can have up to eight trunk groups. Each trunk group can include up to eight ports. For example, you could have four trunk groups of six ports each or three trunk groups of eight ports each.

In the example shown in Figure 1-14, if you have three ATXs (A, B, and C), you could connect them using a single Ethernet segment. However, that would limit the interconnection to 10 Mbps.

To solve this problem, you could connect **A to B** with one trunk group, and connect **B to C** with a second trunk group.

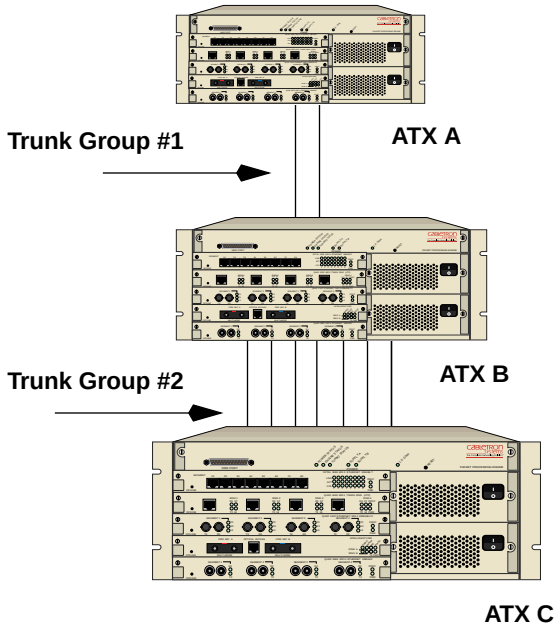


Figure 1-14. Trunk Groups

1.10 LOCAL CONSOLE MANAGER

The Local Console Manager (LCM) is a tool for monitoring, managing, and configuring the ATX, through an out-of-band RS-232 connection attached to a VT-100 type terminal. LCM provides a tool for basic configuration; it is a simple command-line interface. You can also use a standard SNMP-based NMS.

The following sections describe LCM command syntax and the basic LCM commands for logging in and out and getting help. LCM commands used for configuring your ATX are described in the configuration chapters.

1.10.1 Command Syntax Conventions

The following conventions apply as you use LCM commands:

- Press the **Return** key to execute a command after you type it in.
- A **port range** is either a single port number, or a list of port numbers separated by commas or hyphens. For example, "3" is port 3; "3, 7" are ports 3 and 7; "3-5" are ports 3, 4, and 5; and "3-5, 7" are ports 3, 4, 5, and 7.
- To quit any command press the Control-C keys (^C or Ctrl-C).
- You can abbreviate any command where there is no ambiguity; if there is ambiguity, LCM responds with an error message. For example: ATX >ex
Error: ambiguous command
- Commands are not case sensitive.
- Any invalid commands or misspellings entered will receive an error message.
- The previous command can be repeated by typing "!!".
- MAC addresses are displayed in little-endian Ethernet bit order, with each octet separated by a colon. For example:

ATX >address 00:40:27:04:1a:0f

MAC addresses can be entered in big endian format (native FDDI and Token Ring format), by separating octets with spaces rather than colons.

- The default values for filtering command field options appear in square brackets [], for example:

Type:[Entry] (Entry/Exit)>

1.10.2 Basic LCM Commands

The basic LCM commands allow you to get help and log out. LCM commands used for configuring your ATX are described in the configuration chapters. When you want to use LCM, begin by pressing the Return key several times to get the LCM prompt (ATX >).

Note: *The LCM prompt (ATX>) does not appear on the screen immediately. Pressing the Return key repeatedly brings up the LCM prompt. RETURN is the default password.*

exit

Logs you out of LCM. (The exit command is functionally equivalent to the quit command.)

help

Displays the menu of available commands. Help can also be displayed by typing a question mark (?). The output from the help command is displayed below:

```
ATX> ?
ES/1 ATX Local Console Manager
addresses display [any] [ADDR [MASK]]           to display learned addresses
arp [{add PORT# MACaddr IPaddr | delete PORT# IPaddr | display}]
                                                to display arp table information
ataddr [<PORT#> <NETRANGE>]                     to set AppleTalk Network addresses
atroutel [<PORT-RANGE> <OPTIONS>]               to set AppleTalk routing methods
atzone [<PORT#> "<NAME>" {on|off} [default]]      to set AppleTalk zones
baud [BAUD-RATE]                                to change the console baud rate
bridge [PORT-RANGE [OPTIONS]]                   to set bridging methods community
                                                to change the password/community name
disable PORT-RANGE                             to disable a set of ports
display {verbose|terse}                         to select the display mode
elan VIFN {addto|remove} PORT# [802.3][802.5] pvccreate/remove PVC based ELAN
enable PORT-RANGE                              to enable a set of ports
erase                                            to erase configuration information
eventdisplay [<#entries>][continuous]          to display event log eventfilter
```

<code>[clear overwrite stopwhenfull][add del][FILTERS]]</code>	
<code>eventtrap {on off}</code>	to set or display event filter
<code>exit or logout</code>	to manage event/SNMP trap mapping
<code>filters {display modify add delete}</code>	to logout
<code>help or ?</code>	to manage port filters
<code>ident</code>	this menu
<code>ipaddr [PORTS {a cl de di} [ADR [MSK]]]</code>	for software version and board IDs
<code>iproute [PORT-RANGE [OPTIONS]]</code>	to set or display IP addresses
<code>ipxaddr [[[PORT#] NETWORK] FRAMING]</code>	to set IP routing methods
<code>ipxroute [PORT-RANGE [OPTIONS]]</code>	to set IPX Network addresses
<code>mirror [remote PORT-RANGE [OPTIONS]]</code>	to set IPX routing methods
<code>nbcache [PORT-RANGE [OPTIONS]]</code>	to configure port mirroring
<code>nbentries [<#entries>]</code>	to set netbios caching on or off
<code>nbname {display delete}[OPTIONS]</code>	to set/display # nb cache entries
<code>nbtimer [<age_timeout>]</code>	to manage netbios cache
<code>offline MODULE#</code>	to set/display nb cache age timeout
<code>online MODULE#</code>	to stop an interface module
<code>ping [-rvsx] HOST [DATASIZE [COUNT]]</code>	to (re)start an interface module
<code>pvc [{disp add del}]PORT# VPI_RX VCI_RX VPI_TX VCI_TX [VIFN PHYSPORT]</code>	to send ICMP ECHO_REQUEST pkts to host
	to manage PVCs
<code>reboot [time SECONDS cancel]</code>	to re-boot the unit after SECONDS
<code>ringspeed [PORT-RANGE] [4 16]</code>	to set or display 802.5 ring speed
<code>route [{add IPaddr GWaddr PORT# [Hops][IPmask] {delete display} IPaddr}]</code>	to display routing table information
<code>srbridge [BRIDGE#]</code>	to set source-routing bride number
<code>srtb [{ip ipx other all} {on off}] [ste are]</code>	to configure SRTB
<code>status [PORTS]</code>	to display unit or port status
<code>sttimer [TIMER-VALUE]</code>	to set or display st age time
<code>srsegment [PORT# [SEGMENT#] [HOP-COUNT#]]</code>	to set source-routing ring number
<code>traceroute [-m MAX_TTL][-q NQUERIES][-w WAIT] HOST_IP [DATA_SIZE]</code>	to print the route pkts take to host
<code>translate [PORT-RANGE [OPTIONS]]</code>	to set multimedia translations
<code>traplog</code>	to display the most recent SNMP
<code>trapstrunk [PORT-RANGE [{on off}]]</code>	to set or display trunking status
<code>workgroup [NAME [{delete PORT-RANGE [INFO]}]]</code>	to set, delete or display

```
Usage: bridge [PORT-RANGE [{off | transparent | sr | srt} [noBPDU]]]
ATX> id
```

```
Software Currently Running: Release ATX 3.3.09
12-Mar-97
Next Bootstrap (2nd bank) : Release ATX 3.3.09
12-Mar-97
Power-up test failures: none
```

```
System Up Time: 3 days, 21:27:05
```

```
PPE Type: ES/1 ATX
```

```
Bus Type: 800 Mbit
```

Slot	Module Type	Part No
Rev	Serial No	
1	Packet Processing Engine	512-0078-003
2X	0002E400BDC9	
2	Quad Ethernet TP+Crossovr	512-0069
00B	0002E4009FE0	
3	Quad Token Ring	512-0049
01F	0002E4004883	
4	FDDI DAS	512-0054
02A	0002E4001BA0	
5	Quad Fast Ethernet TX	512-0090-002
0X	0002E400665C	
6	Vacant	

```
Usage: ipxaddr [PORT#] IPX-NETWORK [FRAMING]
Where FRAMING is one of:
default ethernet2 llc8022 ethernet8023 ppp
rawfddi snap
```

```
Usage: mirror PORT-RANGE {off | to {PORT# {discard |
truncate} | IPADDR}
mirror remote {off | to PORT# {discard |
truncate}}
```

Usage: nbcache [PORT-RANGE [{off | on}]]

Usage: nbname {display|delete} [big] {<NB_NAME>|any}

Usage: ping [-rvsx] host [datasize [count]]

-r = record route

-v = verbose

-s = send one packet per second continuously

-x = send packets continuously w/o delay

Usage: pvc [{disp|add|delete|clearall}] PORT# VPI_RX
VCI_RX VPI_TX VCI_TX [VIFN|PHYSPORT]

Usage: srtb [{ip|ipx|other|all} {on|off}] [ste|are]

Usage: translate [PORT-RANGE

[{arp|bootp|srArp|ipx|ipxsr|apple|none|netbios|sna|all}
OPTION]]

Usage: translate PORT-RANGE arp {off | oneto6swap}

Usage: translate PORT-RANGE bootp {off | oneto6swap}

Usage: translate PORT-RANGE srArp {off | passRif |
stripRif | passBoth}

Usage: translate PORT-RANGE ipx {off | on | ethernet8023 |ethernet2 |
ethernet8022 | snap}

Usage: translate PORT-RANGE ipxsr {off | passRif |
stripRif | passBoth}

Usage: translate PORT-RANGE apple {off | on}

Usage: translate PORT-RANGE netbios {off | passRif |
stripRif | passBoth}

Usage: translate PORT-RANGE sna {off | passRif | stripRif | passBoth |
onewaybitswap}

Usage: translate PORT-RANGE all {off | passRif |
stripRif | passBoth}

ES/1 ATX> workg

Usage: workgroup [NAME [{delete | PORT-RANGE [INFO]}]]
INFO: {all | ip IP-ADDRESS [NETMASK] | ipx
[IPX-NETWORK]}

ES/1 ATX> trans

Usage: translate [PORT-RANGE

```
[{arp|bootp|srArp|ipx|ipxsr|apple|none|netbios|sna|all}  
OPTION]]
```

```
Port 2 is not configured for token ring.  
Port 3 is not configured for token ring.  
Port 4 is not configured for token ring.  
Port 5 is not configured for token ring.  
Port 6: no translations.  
Port 7 multimedia translations:  
    sna passBoth  
Port 8 multimedia translations:  
    arp oneto6swap  
Port 9 multimedia translations:  
    arp oneto6swap  
Port 10 is not configured for token ring.  
Port 11 is not configured for token ring.  
Port 12 is not configured for token ring.  
Port 13 is not configured for token ring.  
Port 14 is not configured for token ring.  
Port 15 is not configured for token ring.
```

logout

The logout command logs you out of LCM. (The logout command is functionally equivalent to the exit command.)

CHAPTER 2

INSTALLING AND CONNECTING TO THE NETWORK

Carefully unpack the ATX from the shipping carton and inspect it for possible damage. If any damage is evident, contact Cabletron Systems Technical Support. You can also order additional modules separately.

The shipping carton contains:

- The ATX chassis.
- Two power cords.
- Documentation – In addition to this manual, the *ATX MIB Reference Guide* and Release Notes are also included.
- Console Cable Kit

Once you have verified that you have received the modules you ordered and an additional power supply if you ordered it, you can mount the ATX, connect the power supply, and verify that the ATX is fully operational. You can then connect the Local Console Manager. To connect the modules to your network, refer to the module manual for the type of module you are connecting.

2.1 ATX FRONT PANEL

The front panel of the ATX and the I/O modules have both LEDs and switches. LEDs indicate activity taking place; the RESET switch allows you to reset the ATX and its modules. Modules are described in more detail in their manuals. You may want to familiarize yourself with the front panels so you are aware of what is taking place. The front panel of the ATX is shown in Figure 2-1; it also shows front panels of some module types.

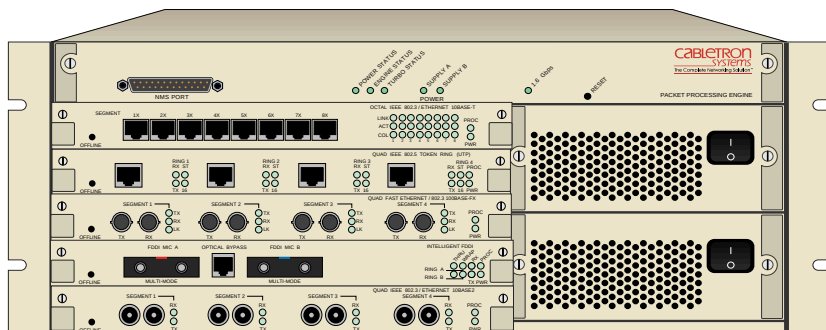


Figure 2-1. ATX Front Panel

ATX LEDs and their functions are described in Table 2-1. Refer to the module documentation for a description of the LEDs for that module.

Table 2-1. Meaning Of ATX LEDs

LED	Meaning
POWER STATUS	On – Power supply is on and the voltage is within the acceptable range.
ENGINE STATUS	On – The Packet Processing Engine (PPE) is operational.
TURBO STATUS	On – The turbo processor is operational.
POWER SUPPLY A	On – Power supply A is generating sufficient voltage for the ATX to operate. Off – The power switch is off, the power supply is not present, or it is malfunctioning.
POWER SUPPLY B	On – Power supply B, the optional power supply is generating sufficient voltage for the ATX to operate. Off – The power switch is off, the power supply is not present, or it is malfunctioning.

ATX switches and their functions are described in Table 2-2. Refer to the module documentation for a description of the switches for that module.

Table 2-2. Description Of ATX Switches

Switch	Function
RESET	Restarts the system software.
	Turns the power supply on or off. The power is on when the rocker switch is on 1. There is a switch for each power supply.

2.2 MOUNTING THE ATX

If the ATX is to be table-mounted, make sure it is within reach of the external power supply and the network cables to which it will be connected. Make sure you allow enough room at the front of the chassis for cable installation and access.

Note: *To ensure adequate ventilation, allow at least 4 inches of space on each side and at the rear of the chassis.*

The ATX may be rack-mounted in a standard 19-inch equipment cabinet (EIA RS-310-C). If you are rack-mounting the ATX, place the chassis in the cabinet and secure it with appropriate fasteners. (Fasteners used to secure an ATX to a rack are not provided with the ATX.) Insert and secure a fastener through each of the four slots in the mounting ears at the front of the ATX chassis as shown below.

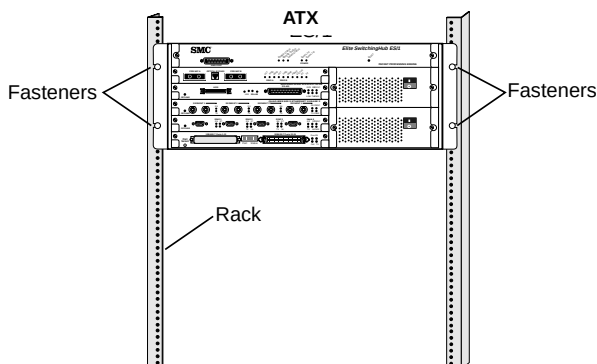


Figure 2-2. Rack-Mounting The ATX

2.3 CONNECTING THE POWER SUPPLY

The ATX utilizes a two power supply system. A primary power source provides the ATX with power. You can purchase an optional, redundant power supply as well. If you use both power supplies, they supply power on a load-sharing basis. If one power supply fails, the remaining power supply automatically provides all the necessary power.

To connect the ATX to an external power source (100 to 120 Vac or 200 to 240 Vac at 47 to 65 Hz), follow the steps below:

1. When using one power supply, plug the power cable into the power socket labeled SUPPLY A on the back of the ATX. When using an optional second power supply, connect its power cable to the SUPPLY B socket on the back of the ATX.
2. Connect the power cable(s) to an outlet.

Note: *It is a good idea to connect the two power supplies to sockets on separate breakers so they will not be dependent on the same power circuit. Using separate circuits allows the ATX to continue to operate if there is a problem with power from one circuit.*

3. Turn on the power supply (or power supplies).

The ATX should now be ready for operation after completing its automatic power-up diagnostics sequence and is connected to the network.

2.3.1 Checking the Power-up Sequence

Before connecting the ATX to any other devices, power on the unit and observe the power-up diagnostics sequence to check for proper operation as described below. The power-up diagnostics sequence completes in approximately 45 seconds depending on the number and type of modules installed.

To check for proper operation, observe the power-up diagnostics. After the power-up sequence is completed, check all LEDs; the status of each LED (off, on, or flashing) should be as shown in Figure 2-3. LEDs are described in Table 2-1. If the status of an LED is not as shown in Figure 2-3 make sure all cables are connected correctly and securely.

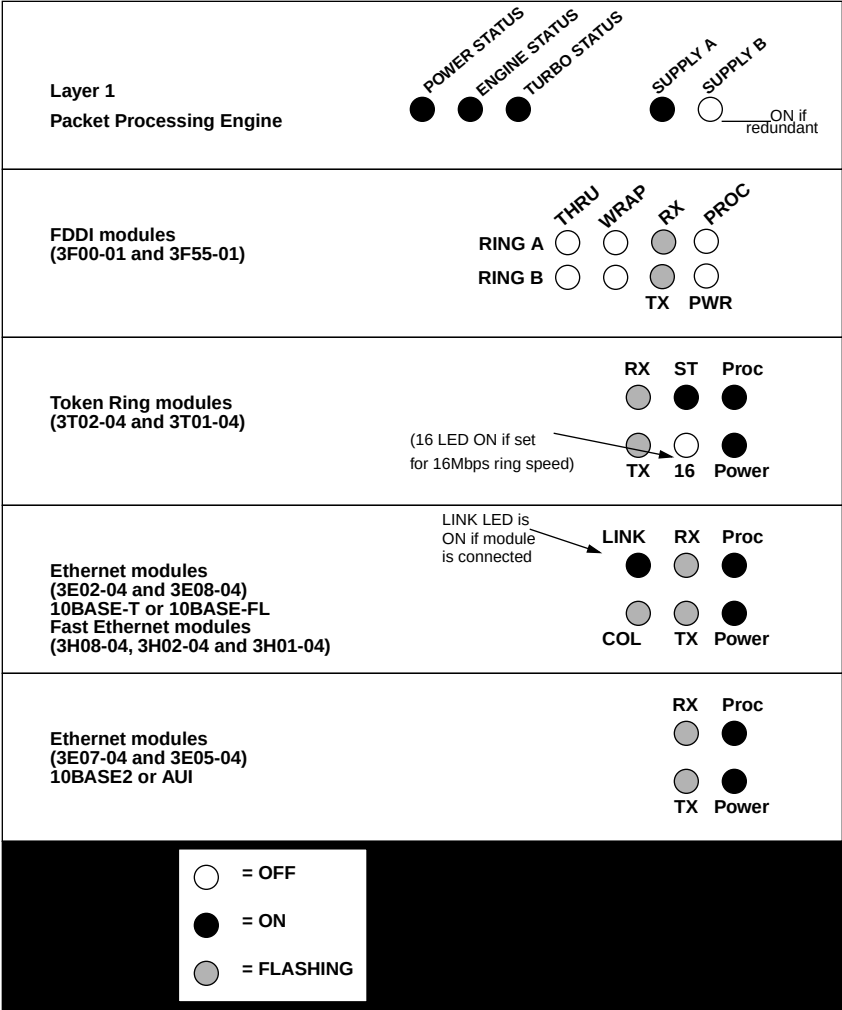


Figure 2-3. LED Activity During Normal Operation

Power-up Diagnostics Sequence

To observe the power-up sequence completely, you may want to repeat it. To restart the power-up sequence, turn the power switch off, then on again, or press the reset button above the power

supplies. LEDs are described in Table 2-1.

When you power up your ATX, the following occurs:

1. All LEDs turn on briefly (this does not apply to all Ethernet Switch models, refer to the *Ethernet Switch Module User Guide*).
2. Individual module LEDs become active, starting with the LEDs on the PPE and continuing downward until all the modules have completed power-up diagnostics.
 - a. The POWER STATUS, ENGINE STATUS, and POWER SUPPLY LEDs on the PPE and the POWER LEDs on the modules are on for approximately 3 seconds.
 - b. The ENGINE STATUS LED on the PPE begins to flash.
 - c. The ENGINE STATUS LED continues to slowly flash while the remaining modules are running power diagnostics.
 - d. The TURBO STATUS LED stays on for approximately 3 seconds; then it flashes.
3. After the last interface module has completed its power-up diagnostics the Packet Processing Engine's ENGINE STATUS LED will stay on solidly.
4. The TURBO STATUS LED will come on, followed by the STATUS or PROC LEDs of the interface modules (from the top down).
5. The LEDs will indicate that the ATX has begun proper operation, as shown in Figure 2-3.

Troubleshooting the Power-up Sequence

If the power-up diagnostics sequence does not proceed as described above, there are some things you can check:

1. Check each interface module to make sure it is fully inserted.

2. Observe the power-up sequence again.
3. If the power-up sequence is still abnormal, contact Cabletron Systems Technical Support, See Chapter 1, **Getting Help**.

Replacing the Power Supply

It is critical that the power supply inserted into the top slot of the ATX chassis be installed very carefully if you are installing it while the ATX is powered on. Failure to use caution while installing the power supply could cause it to come in contact with the bottom of the Packet Processing Engine (PPE) board, causing the PPE to short circuit.

The power supply must slide straight into the chassis underneath the tabs shown in black in Figure 2-4 and rest on its metal support bracket, also shown in Figure 2-4.

Caution: *If you attempt to slide the power supply into the chassis at an angle, or if you position the power supply above the tabs shown in Figure 2-4, you risk short circuiting the PPE board.*

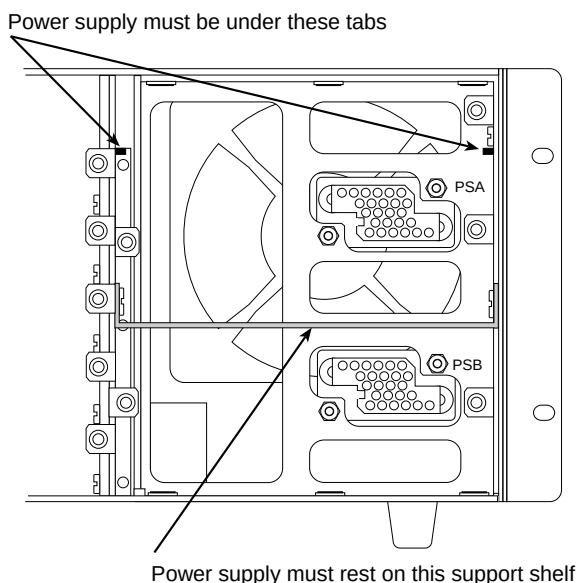


Figure 2-4. Chassis With Power Supply A Positioning Tabs And Supporting Shelf Indicated

To replace the power supply in slot A (the top slot)

1. Turn power switch on Power Supply A (PSA) off.
2. Remove the two thumb screws holding the power supply in place.
3. Pull the power supply straight out.
4. Slide the new power supply straight into the chassis under the tabs shown in Figure 2-4.

The power supply should be placed as shown by the dotted line rectangle in Figure 2-5.

5. Tighten the two screws that hold the power supply into the chassis.
6. Turn the PSA power switch on.

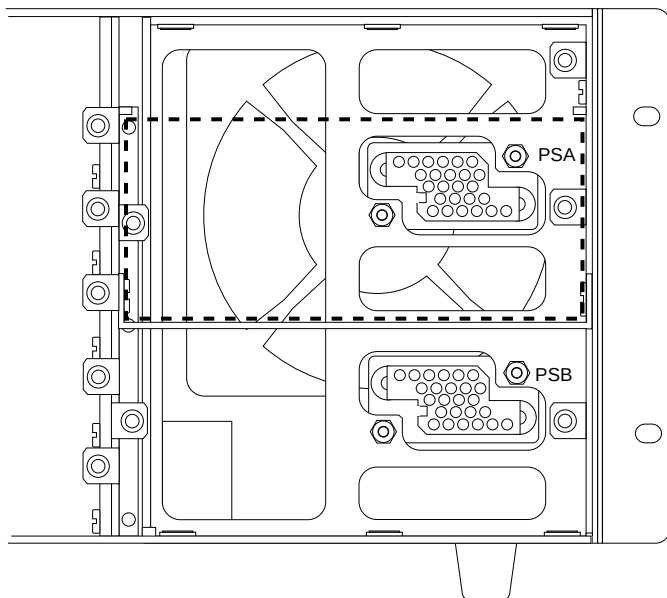


Figure 2-5. ATX With Power Supply A Position Indicated

2.4 CONNECTING THE LOCAL CONSOLE MANAGER

The Local Console Manager is a tool for configuring, monitoring, and managing the ATX through an out-of-band RS-232 connection.

To connect LCM:

1. Attach a null modem at either the terminal end or the ATX port end.

The null modem cable should be a female DB-25 cable. Pinout information is listed in Appendix C, **Null Modem Cable Pinouts**.

2. Connect your ASCII terminal or terminal emulator to the out-of-band management RS-232 port on the ATX.

3. Set the terminal to 9600 baud, 8 data bits, 1 stop bit, and no parity.
4. Press the Return key a few times. If the ATX is powered on, it will respond with its prompt ATX>.

LCM is now ready to use.

Refer to Chapter 1, **Local Console Manager** for a general overview of LCM and the command syntax. Commands for configuring, monitoring and managing, and filters are provided in the chapters dealing with those topics.

CHAPTER 3

CONFIGURING

The ATX does not require any additional configuration to operate as a standard transparent bridge. However, if you want it to communicate with an SNMP manager, you have to assign an IP address to the port through which you will be communicating with the SNMP manager. If you want the ATX to perform IP, IPX routing, or AppleTalk, you need to do some configuring. If you are using a Token Ring module (3T02-04 or 3T01-04) you may need to make some configuration changes as well, refer to the *Token Ring Switch Module User Guide*.

You can configure your ATX using the LCM, which allows you to monitor, manage, and configure your ATX through an out-of-band RS-232 connection. You can also use any SNMP-based network management station.

Configuration parameters are stored in an SNMP standard Management Information Base (MIB), which includes Standard Microsystems' enterprise extensions (variables specific to the ATX). All ATX MIB variables are listed and described in the *ATX MIB Reference Guide*.

This manual provides LCM commands you can use to configure your ATX. If you are using a tool other than LCM, refer to its accompanying documentation.

3.1 CONFIGURING BRIDGING

A bridge is a device that makes it possible to link two or more networks together. Figure 3-1 shows a typical bridging application in which three bridges are used to connect three local area networks (LANs) to a fiber optic “backbone” network. Bridges make interconnected network segments look and function like a single network while reducing intersegment traffic.

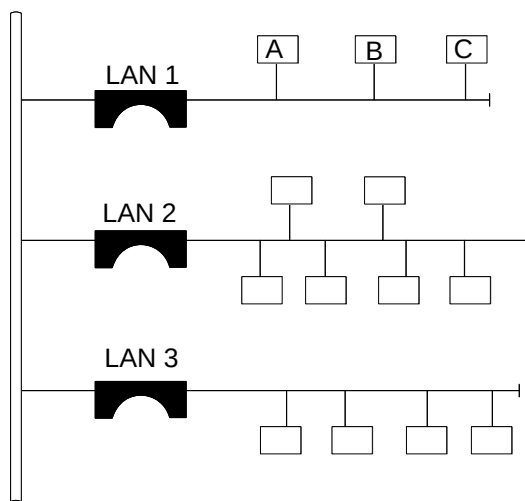


Figure 3-1. Typical Bridging Application

Bridges regulate network traffic on the basis of the source and destination addresses that are in each data packet. Bridges are protocol-transparent, meaning they can handle different types of traffic regardless of the network protocol, for example, IP and IPX. A bridge reads the source and destination address of every data packet it receives and from this information determines where to send the packet.

An important capability of a bridge is its ability to recognize and ignore local traffic. Local traffic refers to data packets that only need to travel within one network segment. For example, a message transmitted from workstation A to workstation C in Figure 3-1 does not need to leave LAN 1. The bridge connected to LAN 1 sees all traffic from LAN 1, including LAN 1 local traffic. But from the source and destination address of each packet, the bridge determines if a packet is local.

If a packet is local, the bridge does not forward it. By forwarding only packets addressed to devices on other segments, bridges reduce unnecessary traffic and thereby enhance the overall performance of the internetwork.

As a bridge reads addresses from the packets it processes, it builds an address table. In this way, it learns the addresses of connected devices. New devices can be added to the network, addresses can be changed, and devices can be removed from the network, without reconfiguring the bridge.

3.1.1 Enabling Bridging Functions

The bridging functions you can enable for the ATX include:

- **Transparent** – End nodes take no part in routing; thus, a transparent bridge places no burden on end nodes.
- **Source Routing** – IBM Source Routing requires source stations to provide routing information within each data frame. Routing intelligence is therefore required at each end node in a Token Ring network.
- **Source Routing Transparent** – A source routing transparent bridge can forward both source routing and transparent data packets. It provides a uniform bridging standard for Ethernet, Token Ring, and FDDI networks.
- **Off** – no bridging at all.

***Note:** If any function other than off is selected, then noBPDU may be specified after the function. A BPDU is a packet the Spanning Tree Protocol uses to communicate with other Spanning Tree compatible devices. Suppressing BPDUs may prevent interoperability problems with bridges that don't conform to the Spanning Tree Protocol. However, it will flood the networks with repetitions of packets if there is a loop. (A loop occurs if a network is connected by multiple bridges back to itself.)*

To use LCM to enable bridging functions for a port or port range, starting from the LCM prompt:

1. Type: **bridge <port range> <functions>**

For example, **bridge 2 srt** would enable source routing

transparent bridging on port 2. LCM responds:

Port 2 bridging: SRT (segment = 1 bridge = 9)

To change the bridging functions for a port, re-issue the bridge command (as described above), using the new option. To set the segment number, use the srsegment command and to set the bridge number use the srbridge command. The default value for the bridge and segment numbers is 0 (zero).

Note: In order to accomplish routing tasks, the ATX must be configured to recognize hexadecimal references. For instance, to route using IPX, a Novell Network Number must be used for configuration purposes. The Novell Network Number, entered in hexadecimal format, identifies the node by the combination of the MAC layer address of its network interface: either its router port or Network Interface Card (NIC). To determine the hexadecimal number from a network number, refer to figure 3-2 below.

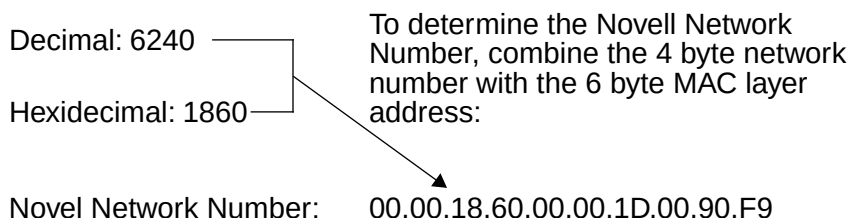


Figure 3-2. Determining a Novell Network Number

3.1.2 Displaying Bridging Functions

To display the bridging functions that are enabled for all ports:

1. Type: **bridge**

LCM responds with a list of all ports and the bridging functions that are enabled. For example:

```
Usage: bridge [PORT-RANGE
[{off|transparent|sr|srt}
```

```
[noBPDU]]]
```

```
Port 2 bridging: Transparent/Translating
Port 3 bridging: SRT (segment = 1 bridge = 9)
Port 4 bridging: SR (segment = 41 bridge = 9)
Port 5 bridging: Transparent/Translating
```

```
.
.
.
```

```
Port 21 bridging: Transparent/Translating
```

You could also type: **bridge <port range>** to look at a specific port or ports. For example **bridge 2-4** would display bridging functions for ports 2, 3, and 4.

3.1.3 Disabling Bridging

To turn off the bridging functions for a port or port range:

1. Type: **bridge <port range> off**

LCM responds, Port 2 bridging: Off

3.2 CONFIGURING IP ROUTING

The ATX is shipped from the factory without an IP address. If you are enabling IP routing, you need to assign addresses to the ports which will be performing routing functions. The LCM command for adding IP addresses is provided in the next section.

3.2.1 Assigning an IP Address

IP addresses for each node must be unique. IP addresses include both a network ID and a node ID; addresses are divided into classes based on what portion of the address is network or node information. The address classes are A, B, and C. (The Network Information Center (NIC) is responsible for assigning IP address).

- Class A addresses are used in very large networks that support many nodes. The first byte identifies the network and the other three bytes identify the node. The first byte of a class A address must be in the range 1-126. The address 100.125.110.10 would identify node 125.110.10 on network 100.
- Class B addresses are used for medium sized networks. The first two bytes identify the network and the last two identify the node. The first byte of a class B address must be in the range 128-191. The address 128.150.50.10 identifies node 50.10 on network 128.150.
- Class C addresses are used for small networks. The first three bytes identify the network and the last byte identifies the node. The first byte of a class C address must be in the range 192-223. The address 192.138.217.10 identifies node 10 on network 192.138.217.
- Class D addresses are used for multicasting. Each multicast group has an unique multicast address. Bits 4-31 identify a particular multicast group. The first 4 bits of a multicast message contain 1110 which identifies the address as a multicast. Multicast addresses range from 224.0.0.0 through 239.255.255.255.

To assign an IP address to a port:

1. Type: **ipaddr <port range> <ip address>**

For example, **ipaddr 6 192.138.217.40** would set the IP address of Port 6 to 192.138.217.40. LCM responds by displaying the IP address table.

3.2.2 Deleting an IP Address

To delete an IP address:

1. Type: **ipaddr <port range> 0.0.0.0 or clear**

LCM responds by redisplaying the IP address table.

3.2.3 Changing a Subnet Mask

You can optionally set the subnet mask for a port. If the subnet mask is 0.0.0.0, the ATX will automatically convert the displayed mask to the standard default, based on the port's IP address class. (Class A address masks are 255.0.0.0, Class B address masks are 255.255.0.0, Class C address masks are 255.255.255.0.)

To change the subnet mask:

1. Type: **ipaddr <port no> <ip addr> <subnet mask>**

For example, **ipaddr 6 192.138.217.40**

255.255.240.0 would set the subnet mask for port 6 to 255.255.240.0. LCM responds by redisplaying the address table.

***Note:** When you change the subnet mask for a port, you must enter the IP address for that port as well. Make sure you enter the IP address for port correctly; whatever you enter becomes the IP address.*

3.2.4 Displaying IP Addresses

The **ipaddr** command displays the IP addresses, subnet masks, and MAC addresses of all ports on the ATX which you are managing.

1. Type: **ipaddr**

Table 3-1. Displaying IP Addresses

Port	IP Address	Address Mask	MAC Address
2	192.138.217.1	255.255.255.0	00:40:27:00:06:1f
3	0.0.0.0	255.0.0.0	00:40:27:00:06:c3
4	192.138.217.10	255.255.255.0	00:40:27:00:06:3e
5	0.0.0.0	255.0.0.0	00:40:27:00:03:7a
6	0.0.0.0	255.0.0.0	00:40:27:00:05:c7
7	192.138.217.20	255.255.255.0	00:40:27:00:04:4a

Table 3-1. Displaying IP Addresses

Port	IP Address	Address Mask	MAC Address
8	192.138.217.50	255.255.255.0	00:40:27:00:06:9e
9	192.138.217.30	255.255.255.0	00:40:27:00:04:b4

3.2.5 Enabling IP Routing Functions

The IP routing functions you can enable for ports on the ATX may be any combination of the following:

- **Off** – no IP routing at all.
- **On** – IP routing, but no inter-router protocols.
- **RIP** – IP routing, with RIP enabled, allows the ATX to send out broadcasts every 30 seconds advertising the networks it knows about, the routes to those networks, and the number of hops to get to there.
- **Proxy** – IP routing, with proxy ARP, allows the ATX to respond to an ARP request on behalf of a device that is located on a network behind it.
- **BOOTP** – Enabling the BOOTP relay option is useful in environments where you have a diskless client and its server is on a network on the other side of the ATX. When the client boots up, it sends out a broadcast requesting the software it needs to download. If BOOTP is not enabled, the ATX won’t forward the broadcast to the network where the server is located. This may also be used to relay DHCP frames.
- **IPM** – Enable IP multicasting. IP multicasting is the transmission of IP packets to a host group. A host group is a set of hosts identified by a single IP address.

To use LCM to enable IP routing functions for a port or port range, starting from the LCM prompt:

Type: **iproute <port range> <functions>**

For example, **iproute 5-6 rip bootp** would enable routing on ports 5 and 6 with the RIP and bootp options on. LCM responds:

Port 5 routing: IP Routing, RIP, Bootp relay
 Port 6 routing: IP Routing, RIP, Bootp relay

3.2.6 Adding an IP Address to a Port

To add an IP address to an ATX port:

Type: **ipaddr <port number> add <ip address>
 <subnet mask> <source route operation option>**

For example, **ipaddr 2 add 192.138.216.111 255.255.255.240** would add 192.138.216.111 and a subnet mask of 255.255.255.240 to port 2. If you do not specify add, the IP address and subnet mask will be added if none already exists. If the port already has an IP address and subnet mask assigned to it, and the “add” command is not used, the new entry will overwrite it.

LCM responds by displaying the updated IP Address entry for the port that changed. To see the full IP Address Table, enter **ipaddr** with no arguments, for example: **ipaddr**

Port	IP Address	Address Mask	MAC Address
2	192.138.217.1	255.255.255.255	00:40:27:00:06:1f
2	192.138.217.2	255.255.255.255	00:40:27:00:06:1f
3	0.0.0.0	255.0.0.0	00:40:27:00:06:c3
4	192.138.217.10	255.255.255.0	00:40:27:00:06:3e
5	0.0.0.0	255.0.0.0	00:40:27:00:03:7a
6	0.0.0.0	255.0.0.0	00:40:27:00:05:c7

Port	IP Address	Address Mask	MAC Address
7	192.138.217.20	255.255.255.0	00:40:27:00:04:4a
8	192.138.217.50	255.255.255.0	00:40:27:00:06:9e
9	192.138.217.30	255.255.255.0	00:40:27:00:04:b4

***Note:** All IP addresses sharing a common subnet must use the same subnet mask. In addition, two IP addresses assigned to the same physical interface must belong to distinct IP subnetworks.*

To add and delete multiple IP addresses on an ATX port, the following subcommands have been added to the LCM **ipaddr** command:

- **add** – Allows you to add an IP address to an ATX port.
- **delete** – Allows you to delete an IP address from an ATX port.
- **clearALL** – Allows you to delete all IP addresses from an ATX port.

3.2.7 Deleting an IP Address From a Port

To delete an IP address from an ATX port.

Type: **ipaddr <port number> delete <ip address>**

For example, **ipaddr 2 delete 192.138.216.111** would delete IP address 192.138.216.111 from port 2. LCM responds by prompting for the next command. To display the current IP Address Table, type **ipaddr** with no arguments.

3.2.8 Clearing All IP Addresses From a Port

To clear all IP addresses from an ATX port:

Type: **ipaddr <port number> clearALL**

For example, **ipaddr 2 clearALL** would delete all of the IP

addresses on port 2. LCM responds by prompting for the next command. To display the current IP Address Table, type **ipaddr** with no arguments.

Note: Before you may issue the *clearAll* command to an ATX port, IP routing must be disabled for that port. To re-enable routing for the port, an IP address must be assigned.

3.2.9 IP Multicast Routing LCM Commands

The **iproute** command displays the IP routing functions enabled for each port. For example:

ATX >**iproute**

```
Usage: iproute [PORT-RANGE [off] [on] [rip]
[ospf] [proxy] [bootp] [ipm]]
Port 2 routing: Network Management Access only
Port 3 routing: Network Management Access only
Port 4 routing: IP Routing, IP Multicasting
Port 5 routing: IP Routing, IP Multicasting
Port 6 Routing: IP Routing, IP Multicasting
Port 7 Routing: Network Management Access only
.
.
.
.
Port 21 routing: IP Routing, Proxy
```

The **iproute <port range>** command displays the IP routing functions that are enabled for the specified ports. For example:

ATX >**iproute 2-4**

```
Port 2 routing: IP routing, RIP
Port 3 routing: IP Routing, Proxy
```

The **iproute <port range> <functions>** command selects the IP routing functions for the specified ports. Those IP routing functions include: **iproute** [PORT-RANGE [off][on][rip][proxy][bootp][ipm]].

3.2.10 Displaying IP Routing Functions

To display the IP routing functions that are enabled for all ports:

Type: **iproute**

LCM responds with a list of all ports and the routing functions that are enabled.

```
Usage: iproute [PORT-RANGE] [off] [on] [rip]
[proxy] [bootp]]
```

```
Port 2 routing: IP Routing, RIP
```

```
Port 3 routing: IP Routing, RIP Bootp relay
```

```
Port 4 routing: IPX
```

```
Port 5 routing: IP Routing, Proxy ARP
```

```
.  
.
.
```

```
Port 21 routing: IP Routing,
```

You could also type: **iproute <port range>** to look at a specific port or ports. For example, **iproute 2-4** would display routing functions for ports 2, 3, and 4.

3.2.11 Disabling Routing Functions

To turn off the IP routing functions for a port or port range:

Type: **iproute <port range> <off>**

LCM responds:

```
Port 2 routing: Off
```

3.3 CONFIGURING IPX ROUTING

If you are enabling IPX routing, you need to assign unique addresses to the ports which will be performing routing functions. The LCM command for assigning IPX addresses is provided in the following section, *3.3.1 Assigning an IPX Address*.

3.3.1 Assigning an IPX Address

IPX addresses for each port must be unique and non-zero. When you assign an address, you can also designate the frame type. Frame types are listed below with the value you enter listed in parenthesis:

- Ethernet 2 (ethernet2)
- Raw 8023, the default for Ethernet (ethernet802.3)
- 8022, the default for Token-Ring and FDDI (llc802.2 or ethernet802.2)
- SNAP (snap)
- Raw FDDI, valid only for FDDI (rawfddi)
- PPP, the only option for HSSI (ppp)

To assign an IPX address to a port:

Type: **ipxaddress <port number> <ipx address>
<framing type>**

For example, **ipxaddr 6 0x12345678** would set the IPX address of Port 6 to 0x12345678. LCM responds by displaying only the changed IPX entries.

You can use a decimal form to enter an address if you don't start the address with 0x.

To change an IPX address:

Type: **ipxaddress <port number> <new address>**

LCM responds by displaying the IPX address table.

3.3.2 Displaying IPX Addresses

The **ipxaddr** command displays the IPX addresses, node ID, and framing type for all ports on the ATX which you are managing.

Type: **ipxaddress**

Table 3-2. Displaying IPX Addresses

Port	IPX Network	Node ID	Framing
2	0x11223344	00:40:27:00:06:1f	Ethernet 802.3
3	0x55667788	00:40:27:00:06:c3	Ethernet 802.3
4	0x99001122	00:40:27:00:06:3e	Ethernet 802.3
5	0x33445566	00:40:27:00:03:7a	LLC 802.2
6	0x12345678	00:40:27:00:05:c7	Ethernet 802.3
7	0x77665544	00:40:27:00:04:4a	LLC 802.2
8	0x31265488	00:40:27:00:06:9e	Ethernet 802.3
9	0x22446688	00:40:27:00:04:b4	Ethernet 802.3

3.3.3 Enabling IPX Routing Functions

The IPX routing functions you can enable for ports on the ATX may be:

- **Off** – no IPX routing at all
- **On** – IPX routing
- **SR** – IPX routing over source routing

To use LCM to enable IPX routing functions for a port or port range, starting from the LCM prompt:

Type: **ipxroute <port range> <functions>**

For example, **ipxroute 5-6** would enable IPX routing on ports 5 and 6. LCM responds:

Port 5 IPX routing: enabled

Port 6 IPX routing: enabled

3.3.4 Displaying IPX Routing Functions

To display the IPX routing functions that are enabled for all ports:

Type: **ipxroute**

LCM responds with a list of all ports and the routing functions that are enabled.

```
Usage: ipxroute [PORT-RANGE [{off | on | sr}]]
```

```
Port 2 IPX routing: enabled
```

```
Port 3 IPX routing: enabled
```

```
Port 4 IPX routing: enabled
```

```
Port 5 IPX routing: enabled
```

```
.
```

```
.
```

```
.
```

```
Port 21 IPX routing: enabled
```

You could also type: **ipxroute <port range>** to look at a specific port or ports. For example, **ipxroute 2-4** would display routing functions for ports 2, 3, and 4.

3.3.5 Disabling IPX Routing

To turn off the IPX routing functions for a port or port range:

Type: **ipxroute <port range> <off>**

LCM responds:

```
Port 2 IPX routing: disabled
```

3.4 CONFIGURING APPLE TALK ROUTING

AppleTalk routing can be enabled on a per port basis using the Local Console Manager (LCM). (For basic LCM commands and command syntax, refer to Chapter 1, **Local Console Manager**) When you enable AppleTalk routing, the first port you enable starts the seed process to enable all AppleTalk routers to acquire

their network number. Refer to Chapter 1, **Appletalk Routing** for a conceptual overview of AppleTalk routing, including the concept of a seed router.

Whenever you enable a port, it goes through the process of acquiring its address again. Once the network has been seeded, a newly enabled port takes its network information from the other routers on the network.

3.4.1 Enabling AppleTalk Routing

To enable AppleTalk routing on a port or port range:

Type: **atrout** <port range> on.

For example, **atrout 4-8 on** would enable AppleTalk routing on ports 4, 5, 6, 7, and 8. LCM responds:

Port 4 AppleTalk routing: enabled

Port 5 AppleTalk routing: enabled

Port 6 AppleTalk routing: enabled

Port 7 AppleTalk routing: enabled

Port 8 AppleTalk routing: enabled

3.4.2 Displaying AppleTalk Routing Functions

You can use the **atrout** command to display the AppleTalk routing state for all ports on the ATX.

To display the AppleTalk routing state for all ports

Type: **atrout**

LCM responds:

Usage: atrout [<port range> {off |on}]

Port 2 AppleTalk routing: disabled

Port 3 AppleTalk routing: disabled

Port 4 AppleTalk routing: enabled

Port 5 AppleTalk routing: enabled

Port 6 AppleTalk routing: enabled

Port 7 AppleTalk routing: enabled

```
Port 8 AppleTalk routing: enabled
.
.
.
Port 21 AppleTalk routing: disabled
```

3.4.3 Disabling AppleTalk Routing

AppleTalk routing can be disabled on a per port basis using LCM. AppleTalk packets that are received on disabled ports are discarded.

To disable AppleTalk routing on a port or port range:

Type: **atrout <port range> off.**

For example, **atrout 4-8 off** would disable AppleTalk routing on ports 4, 5, 6, 7, and 8. LCM responds:

```
Port 4 AppleTalk routing: disabled
Port 5 AppleTalk routing: disabled
Port 6 AppleTalk routing: disabled
Port 7 AppleTalk routing: disabled
Port 8 AppleTalk routing: disabled
```

3.4.4 Assigning a Network Number

When you want a port to “seed the network”, that is, to be the port that all other ports learn their network number from, you must manually assign the port its network number. In AppleTalk Phase 2, the network number is actually a range of numbers. (A range of 0-0 indicates that this port is not the seed port.)

Once the last network has been seeded, when a port is enabled, it learns its network number from the other routers on the network. The only way to reseed the network is to bring down all the routers at once and then start the seed process again. Any port can act as the seed router, once it has seeded the network it is no different from any other router on the network. If it goes down, newly enabled ports will continue to learn their network identity from

the previously seeded information.

You can create a new network range by using the **ataddr** command to assign a new range. However, if the network has already been seeded, the number you assign will not be used; the seeded information takes precedence.

To assign a network number:

Type: **ataddr <port number> <start range> - <end range>**

For example, **ataddr 4 5-10** would create the network number range 5-10 on port 4. LCM responds:

Port	CFG-Range	Active Range	DDP-Addr	Net-Cfg	Zone-Cf
4	5-10	0-0	0.0	unconfigured	unconfigured

After the network has processed the operation, the LCM may display:

Port	CFG-Range	Active Range	DDP-Addr	Net-Cfg	Zone-Cfg
4	5-10	5-10	5.2	configured	configured

Decimal numbers with no leading zeroes are used for values.

Notes: *If there are no other devices on the network, the network configuration (Net-Cfg) and zone configuration (Zone-Cfg) status is listed as unconfigured. As soon as another device comes up, the ATX configures itself and the status is changed to configured. The configuration range (Cfg-Range) is the network number range you have assigned to this port. The active range is the network number that was seeded to your network. If the network has already been seeded, even if you assign a network number to a port, the port still uses the seeded information. To change the active range, bring all routers on the network down and reseed the network with the new range you want to use.*

3.4.5 Displaying the Network Number

You can find the current network range for any port on which AppleTalk routing is enabled by using LCM.

To find the network number for a port:

Type: **ataddr**

LCM responds:

Usage: ataddr [<port number> <net-range>]

Port	CFG-Range	Active Range	DDP-Addr	Net-Cfg	Zone-Cfg
2	0-0	0-0	0.0	unconfigured	unconfigured
3	300-400	298-400	300.2	configured	configured
4	100-200	100-200	300.2	garnered	garnered

Notes: *If there are no other devices on the network, the network configuration (Net-Cfg) and zone configuration (Zone-Cfg) status is listed as unconfigured. As soon as another device comes up, the ATX configures itself and the status is changed to configured. When the configured zone and network status is listed as garnered, it means that this port learned its network number and zone name from the seed router.*

3.4.6 Adding a Zone Name

Every AppleTalk device belongs to a zone, which is a logical grouping of devices. For example, you may want to create zones that correspond to department names such as Engineering, Marketing, or Sales. Zone names may have as many 32 characters and they may include spaces. You must enter the zone name in quotes for it to be recognized.

To add a zone name.

Type: **atzone <port number> <"zone name"> on.**

For example, **atzone 6 "Engineering" on**, would create the zone name Engineering on port 6. LCM responds:

AppleTalk Zones

Port 6 Engineering

To make the zone name you are adding the designated default zone name:

Type: **atzone <port number> <"zone name"> on default.**

For example, **atzone 6 "Engineering" on default**, would create the default zone name Engineering on port 6. LCM responds:

AppleTalk Zones

Port 6
(default)Engineering

3.4.7 Displaying a Zone Name

You can use LCM to display AppleTalk zone names that are currently assigned. To display zone names:

Type: **atzone**

LCM responds:

Usage: **atzone [<Port number> "<Name>" {on|off} [default]]**

AppleTalk Zones

Port 6
Engineering

3.5 CONFIGURING TRUNKING

If your network configuration requires you to connect two or more ATXs together, but the applications you are running over the network require more than a single LAN connection (10 Mbps of bandwidth for Ethernet) between ATXs, you can use the built-in trunking feature to raise bandwidth up to 8 times the single LAN connection bandwidth (80 Mbps for Ethernet), without installing

additional hardware on your network. You can use LCM to configure trunking. You can enable trunking between ATXs or between an ATX and a Fast Network 10. For more information on trunking, see section, *1.9 Trunking*.

3.5.1 Enabling Trunking

To enable trunking you would:

1. Connect the desired ports of the ATXs together using the appropriate cables.

ATX A is handling only a small traffic load. Therefore, the **A to B** trunk group has just two ports per ATX. ATXs B and C are expected to support a higher traffic load. Therefore, the **B to C** group has eight ports.

2. Using LCM, turn on trunking for the connected ports on each ATX.

For ATX A, at the LCM prompt:

Type: **trunk 2,3 on**

For ATX B, at the LCM prompt:

Type: **trunk 3-10, 14-15 on**

For ATX C, at the LCM prompt:

Type: **trunk 3-10 on**

Each ATX automatically determines which ports are part of which trunk group. After trunk group configuration, the ATXs complete the standard 802.1D Spanning Tree state changes, treating each trunk group as a single 802.1D Spanning Tree Port.

802.1D Spanning Tree could take up to thirty seconds to resolve which ATX ports are to become forwarding ports. As ports within a trunk group become forwarding ports, traffic within the trunk

group is momentarily halted to guarantee the first-in, first-out ordering of the Ethernet packets.

Note: *The ATX-to-ATX connections must be point-to-point. There cannot be any other devices on those LAN segments. The ports used for trunking can be in any order. However, both ends of the ATX-to-ATX connections must have trunking enabled for the ports that are being used for the connections.*

3.5.2 Disabling Trunking

Disabling trunking on a port causes that port to return to standard Spanning Tree operation. You must disable trunking on both ends of the point-to-point connection, otherwise bridging is not enabled.

To turn off trunking, at the LCM prompt:

Type: **trunk <port-range> off**

For example, **trunk 2-4 off**

3.6 CONFIGURING MULTICAST STORM PROTECTION

The ATX provides automatic protection against multicast storms. Multicast storms are excessive broadcasts to all stations, typically caused by a malfunctioning device. They can result in severe network performance problems, including causing the network to crash.

The way you protect against multicast storms is to define an acceptable rate for multicast traffic across a port. In many ways this feature is similar to filtering, however, multicast storm protection does not involve the use of filters.

Each ATX port can be individually configured for automatic multicast storm protection. You define what level of multicasts the ATX will recognize as a multicast storm, by specifying the number of multicast packets that may be transmitted with a given time period.

For example, if you configure port 3 to accept no more than 5 multicasts per 60 seconds, any multicasts destined for port 3 are discarded after the first 5. After 60 seconds have elapsed, another 5 multicasts to port 3 will be allowed. This maintains an effective maximum rate of 5 multicast packets per minute.

3.7 MODIFYING MIB VARIABLES

Specific instructions for controlling ATX operations, modifying parameters, etc., depend on the NMS you are using. This manual provides instructions for using LCM commands, but LCM commands don't exist for all configuration options. You may need to modify your configuration using an NMS. This section provides a list of common MIB variables you may want to change. (Refer to the *ATX MIB Reference Guide* for a complete listing and description of MIB variables.)

Each variable is first described in words and is then identified in MIB form, for example, `configGetPass - {config 3}`. The `DisplayString` line shows the range of values that may be used for the given parameter. In each case, the `DisplayString` is a string of ASCII characters.

3.7.1 System Contact

The system contact parameter identifies a contact person who is responsible for operation of the ATX. Typically this parameter includes the person's name, company or division name, and telephone number.

```
sysContact - {system 4}  
DisplayString (SIZE (0..255))
```

3.7.2 System Name

The system name is a name assigned to the ATX by the network administrator. By convention, the system name is the fully qualified domain name. (This name then becomes the LCM prompt.)

```
sysName - {system 5}  
DisplayString (SIZE (0..255))
```

3.7.3 System Location

The system location identifies the physical location of the ATX.

```
sysLocation - {system 6}  
DisplayString (SIZE (0..255))
```

3.7.4 Authentication Password

The set password and get password variables (from the Cabletron MIB Configuration Status Group) must be initialized with the correct authentication passwords.

All requests from any SNMP manager contain a community name field. For set requests, the community name must match the set password; otherwise, the request will be rejected by the ATX. For get requests, the community name must match either the set password or the get password.

Set Password

The set password variable (configAnyPass) must be set to the value of the community name used by the SNMP manager for performing either set or get operations. A zero length password means that any community name is acceptable.

```
configAnyPass - {config 2}  
DisplayString (SIZE (0..24))
```

This variable is also used as the login password for LCM.

Note: *configAnyPass* permits read-write access. *configGetPass* permits read only access.

Get Password

The get password variable (*configGetPass*) must be set to the value of the community name used by the SNMP manager for performing get operations. A zero length password means that any community name is acceptable.

```
configGetPass - {config 3}  
DisplayString (SIZE (0..24))
```

Aging Parameter

Dynamic (learned) addresses are automatically deleted from the ATX address table after a certain length of time. The aging time default is 5 minutes as set by the IEEE 802.1d standard. However, the aging parameter can be changed, using the MIB variable *dot1dTpAgingTime*.

The ATX continually compares the actual age of each dynamic address against the age specified by the *dot1dTpAgingTime* parameter and deletes any addresses that are older than the age specified (or older than 5 minutes if you are using the default). Since most communication takes place within a very short period of time, the aging parameter can usually be set for a relatively short time.

Note: *Static addresses (those added by the user) are not aged.*

Traps (acknowledge)

The ATX can be configured to retransmit traps (alarms) until the traps are acknowledged by the NMS.

```
sysTrapAck, sysTrapTime, sysTrapRetry
```

Configuration Alarm Dynamic

When the ATX learns a new address or ages (deletes) an old address it may or may not send a trap based on the value of this variable.

```
configAlarmDynamic, addrAlarmMAC
```

3.8 CONFIGURING NETBIOS NAME CACHING

The Netbios name caching function initially comes up disabled. To enable or disable name caching, the ports to enable must be provided. If you enable a port for Netbios Name Caching, you've turned on the capability to learn the netbios names coming from that port. If a port is disabled for caching, it will not prevent a broadcast from going out that port. If a port is disabled for caching, it will not prevent a broadcast from going out that port. If you have netbios machines on two separate ports, both ports must be enabled for Netbios cache for the functionality to work. To display which ports are enabled or disabled for Netbios Name Caching, simply type 'nbcache' without arguments or the 'bridge' command. To display the status of a specific port, type:

```
nbcache <port range> {on|off}
```

The `nbentries` command with no arguments will display the current number of entries which can possibly be saved in the cache. The number of entries can be modified by providing the argument `nbr_entries`. Since memory is allocated at boot time, one would need to reboot to get more/less space. The default value is 512, whereas the maximum number of entries is 5000.

```
nbentries [nbr_entries]
```

Note: *Changing the number of entries directly affects the performance of NetBIOS Name caching. Keeping the number of names to the minimum amount necessary ensures peak performance for the ATX.*

The `nbtimer` command with no arguments displays the current

value of the Netbios aging timer. The age-timeout argument can be modified and is interpreted in terms of seconds. This timer is the amount of time a Netbios name remains in cache without activity. The default will be the same as that for spanning tree which is 5 minutes or 300 seconds. To empty out all entries from cache, one can set the timeout to zero. The default value is 300 seconds.

```
nbtimer [age_timeout]
```

The nbname command requires at least one argument. If the display option with the ANY argument is selected, all current entries in cache will be displayed. If the display option with the [nbname] option is selected, only that entry matching the Netbios name is displayed. If the delete option is selected then provide the Netbios name which you'd like to delete. To delete all entries from cache, use the nbtimer command.

```
nbname  
{display[nb_name|big|any]|delete{nb_name|any}}
```

3.9 VIRTUAL WORKGROUP LCM COMMANDS

The LCM command format is:

workgroup name ports type info

name 1-16 characters; identifies the workgroup

ports range of ports separated by (-) or (,)

type ALL or IP or IPX

info ip address\mask or ipx network number (hex or decimal);
NA for type ALL

Examples:

```
workgroup eng 3-7 all
```

```
workgroup sales 10,11,12,13 ip 134.141.141.0  
255.255.255.0
```

```
workgroup mktg 11,12-18 ipx 0x1234
```

3.10 CLASSIFICATION

When a broadcast packet is received on a workgroup defined port, the packet is classified as being IP (IP, ARP or RARP), IPX(SAP, RIP, SPX or NCP) or ALL (any protocol type). Based on this classification, the broadcast will only be forwarded to the ports within that workgroup. If there is no workgroup defined for the receiving port the broadcast is forwarded out all other ports regardless of the exiting port's workgroup configuration.

3.10.1 Workgroup of Type ALL

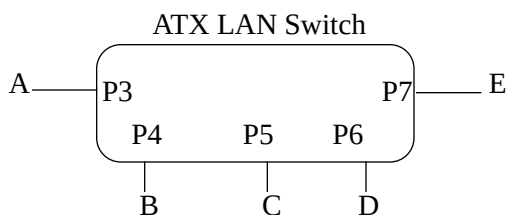
When a broadcast of any protocol type is received by a port with only an ALL workgroup defined, the packet will be broadcast out every port in the ALL workgroup (see Example #1).

Example #1

Defined workgroups:

```
workgroup red 3-5 ALL
```

```
workgroup blue 5-6 ALL
```



Broadcast from A will only be seen by B and C

Broadcast from B will only be seen by A and C

Broadcast from C will only be seen by A, B and D

Broadcast from D will only be seen by C

Broadcast from E will be seen by all forwarding ports

3.10.2 Workgroup of Type IP

The destination IP address within the broadcast packet is used to determine the workgroup (see Example #2). This IP address is matched against the IP network address and IP network mask defined in the workgroup for the receiving port. If the destination IP address does not match the IP workgroup defined the packet is forwarded out all other ports. If the destination IP address is a broadcast address, the source IP address is used to determine the correct workgroup. If there is no destination IP address (i.e. RARP), then the packet is forwarded out all of the IP workgroups for the receiving port. If the packet is an IP multicast but not broadcast (i.e. class D address) the workgroups are ignored and the normal forwarding criteria is applied.

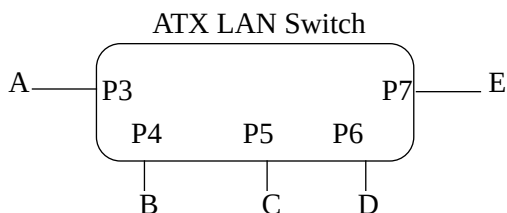
Example #2

Defined workgroups:

```
workgroup red 3-5 All
```

```
workgroup blue 5-6 IP 100.100.1.0  
255.255.255.0
```

```
workgroup green 6-7 IP 100.100.2.0  
255.255.255.0
```



An ARP from:

A or B destined for 100.100.1.xxx will only be seen by A, B and C

A or B destined for 100.100.2.xxx will only be seen by A, B and C

A or B destined for 100.100.3.xxx will only be seen by A, B and C

C destined for 100.100.1.xxx will only be seen by D

C destined for 100.100.2.xxx will be seen by all forwarding ports

C destined for 100.100.3.xxx will be seen by all forwarding ports

D destined for 100.100.1.xxx will only be seen by C

D destined for 100.100.2.xxx will only be seen by E

D destined for 100.100.3.xxx will be seen by all forwarding ports

E destined for 100.100.1.xxx will be seen by all forwarding ports

E destined for 100.100.2.xxx will only be seen by D

E destined for 100.100.3.xxx will be seen by all forwarding ports

3.10.3 Workgroup of Type IPX

To determine the workgroup of an IPX broadcast the destination IPX network number is used (see Example #3). If the destination IPX network number is zero, the packet is forwarded out all of the IPX workgroups for the receiving port. If the broadcast has a non-zero IPX network number, there are a few possibilities. The IPX workgroup with the same IPX network number is used. If the destination IPX network number does not match the workgroup defined and a default IPX workgroup (IPX network number 0) is defined that workgroup is used (see Example #4). If destination IPX network number does not match the defined workgroup and there is no default IPX workgroup, the packet is forwarded out all other forwarding ports.

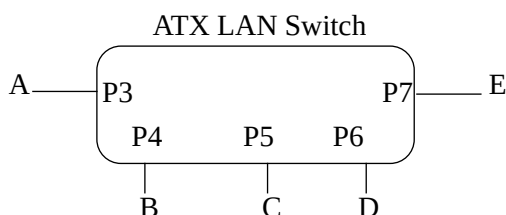
Example #3

Defined workgroups:

```
workgroup red 3-5 all
```

```
workgroup blue 5-6 ipx 0x1234
```

```
workgroup green 7 ipx 0x999
```



A SAP from:

A or B destined for the 0x1234 network will only be seen by A, B and C

A or B destined for the 0x999 network will only be seen by A, B and C

A or B destined for the 0x000 network will only be seen by A, B and C

C destined for the 0x1234 network will only be seen by D

C destined for the 0x999 network will be seen by all forwarding ports

C destined for the 0x000 network will only be seen by D

D destined for the 0x1234 network will only be seen by C

D destined for the 0x999 network will be seen by all forwarding ports

D destined for the 0x000 network will only be seen by C

E destined for the 0x1234 network will be seen by all forwarding ports

E destined for the 0x999 network will stay local to E

E destined for the 0x000 network will stay local to E

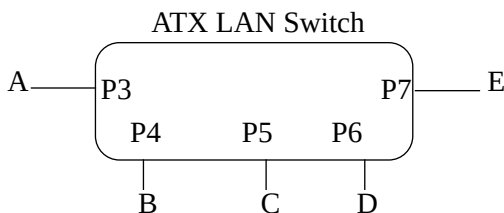
Example #4

Defined workgroups:

workgroup red 3-5 all

workgroup blue 5,6,7 ipx 0

workgroup green 7 ipx 0x999



A SAP from:

A or B destined for the 0x1234 network will only be seen by A, B and C

A or B destined for the 0x999 network will only be seen by A, B and C

A or B destined for the 0x000 network will only be seen by A, B and C

C destined for the 0x1234 network will only be seen by D and E

C destined for the 0x999 network will only be seen by D and E

C destined for the 0x000 network will only be seen by D and E

D destined for the 0x1234 network will only be seen by C and E

D destined for the 0x999 network will only be seen by C and E

D destined for the 0x000 network will only be seen by C and E

E destined for the 0x1234 network will only be seen by C and D

E destined for the 0x999 network will stay local to E

E destined for the 0x000 network will only be seen by C and D

3.10.4 Same Port in Multiple Workgroups

In the event that a port is defined in workgroups of ALL and IP or IPX, the forwarding criteria for IP packets or IPX packets will be based on IP workgroup or IPX workgroup respectively. If the IP\IPX broadcast does not match the IP or IPX workgroup the packet will be forwarded out every other port. It will NOT revert back to the criteria set for the ALL workgroup defined on that port. For instance, in Example #2 port 5 is a member of two workgroups, RED of type ALL and BLUE of type IP. When station

C sends an IP packet destined for any network other than 100.100.1.0 the broadcast is forwarded out every other forwarding port. Even though port 5 is a member of two workgroups it does not fall back to the RED workgroup's criteria.

3.10.5 Workgroup to Workgroup Communication

This type of communication can only be achieved by routing. With the ATX LAN Switch having the ability to route IP packets, it will route between IP workgroups (See Example #5). However, the ATX LAN Switch will NOT be able to route between IPX workgroups. The reason is that the ATX LAN Switch does not have the ability to enable IPX routing on multiple ports with the SAME IPX network number. Therefore, communication between IPX and ALL workgroups can only be achieved via an external router.

Example #5

Defined workgroups:

```
workgroup red 3-5 ip 134.141.100.0  
255.255.255.0
```

```
workgroup blue 6-7 ip 134.141.200.0  
255.255.255.0
```

IP Configuration with IP enabled on all ports:

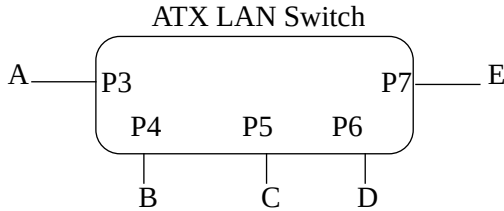
```
ipaddress P3 134.141.100.3 255.255.255.0
```

```
ipaddress P4 134.141.100.4 255.255.255.0
```

```
ipaddress P5 134.141.100.5 255.255.255.0
```

```
ipaddress P6 134.141.200.6 255.255.255.0
```

ipaddress P7 134.141.200.7 255.255.255.0



Results:

- Stations A, B and C IP communication will be switched between ports 3, 4 and 5 since they are on the same subnet of 100.
- Stations D and E IP communication will be switched between ports 6 and 7.
- If A, B or C needs to communicate with D or E and vice versa. The receiving port will have the ability to route the packet to the 200 or 100 subnet respectively since routing is enabled on all ports.

3.11 LOCAL AND REMOTE PORT MIRRORING COMMANDS

The LCM command format for Local Port Mirroring is:

mirror port-range off

port-range - range of mirrored ports

off - to turn local port mirroring off on the ports specified

mirror port-range to port# oversize

port-range - range of mirrored ports

port# - the diagnostic port on the local ATX

oversize - discard or truncate; what to do with oversized packets

The LCM command format for Remote Port Mirroring is:

Local ATX (in reference to the diagnostic port)

mirror remote off

off - to turn remote port mirroring off

mirror remote to port# oversized

port# - the diagnostic port on the local ATX

oversized - discard or truncate; what to do with oversized packets

Remote ATX (in reference to the diagnostic port)

mirror port-range off

off - to turn remote port mirroring off

mirror port-range to Ipaddr

port-range - range of mirrored ports on remote ATX

Ipaddr - ip address of the local ATX where the diagnostic port resides

***Note:** Both ATX LAN Switch's have to have port mirroring turned off in order to fully disable the remote port mirroring function.*

3.11.1 Types of Media and Framing

Mirrored and diagnostic ports have no restrictions and can be any of the ATX LAN Switch's interfaces, Token Ring, Ethernet, Fast Ethernet or FDDI. However, it is recommended that the diagnostic port and mirrored ports are of the same media type and framing. This is because in an intermixed mode, due to the differences in the physical layers, mirrored packets may be translated or dropped. For example, when an 802.5 packet is mirrored out to an 802.3 interface, the MAC addresses are translated (big endian to little endian) and the length field is added to the original frame.

Furthermore, mirroring traffic of a higher speed interface out to a lower speed interface may impose a strain on performance (e.g. capturing FDDI traffics to a 4 Mbps Token Ring). When the size of the mirrored packet exceeds the size of the maximum transport unit (MTU) of the diagnostic port, the packet is labeled as oversized. As an option for local mirroring in an intermixed mode, the ATX can be configured to truncate or discard oversized packets.

3.11.2 Packet Capturing and Mirroring

The mirroring of network traffic is performed by the ATX LAN Switch software, and the mirror image reflects the ATX LAN Switch internal representation of the packets. Certain physical layer information (such as Access Control and Frame Control in 802.5 frames) will not be available. The difference in the physical layers are minor, and should not impair the normal usage of the port mirroring as it is mostly used in MAC and network layers.

The ATX LAN Switch mirror software attempts to minimize any differences between the internal and external formats when the frame is mirrored out the diagnostic port. Other than the possible framing translation, MAC layer should have only minor or no differences between the mirrored image and the raw frame on the wire. On the network layer, there should be no alteration. For example, when an inbound routed packet is mirrored, the image reflects the packet prior to any changes made by the ATX LAN Switch routing software.

The ATX LAN Switch mirror software maintains the original packet ordering of bridging frames between the inbound and outbound interfaces. The bridging packets include the Transparent, Source Routed and Transparent Source Routed frames. Network layer routing traffic is not subject to this requirement, and the sequence of routed packets may occasionally be out of order (as in the cases without port mirroring).

3.11.3 Mirrored Filters

The ATX also allows you (via the existing port filtering feature; Chapter 5 in the ATX LAN Switch User’s Guide) to establish “mirror filters” which can help reduce the amount of traffic seen by the diagnostic port. Using a “mirror filter,” you can restrict the amount of monitored traffic by filtering inbound or outbound packets according to source and destination addresses, packet types, frame protocols and offsets within the data field.

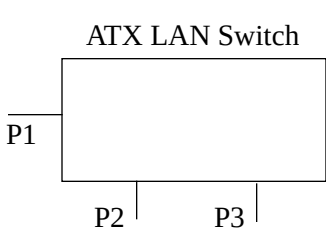
In port filters, there are currently two types you can select from: Entry and Exit. With the addition of port mirroring, there are now four types: Entry, Exit, PMEntry and PMExit. PMEntry applies to any packet entering the port and PMExit is any packet leaving the port. See Configuration Examples for implementation. The rest of the parameters for setting up filters are identical, independent of what the type is.

There are two major differences between mirror filter and packet filter:

- A mirrored filter has the exact opposite affect as a port filter. Mirrored filters will pass the traffic matching the filter rather than being blocked as in packet filtering.
- Both inbound packets to the ATX and outbound packets generated by the ATX are subject to the mirror filtering.

3.11.4 Example #1: LOCAL Port Mirroring

Port 1 is the diagnostic port where the analyzer resides.



Ports 2 and 3 are the mirrored ports

mirror 2-3 to 1
discard

or

mirror 2-3 to 1
truncate

Mirror Filters with LOCAL Port Mirroring:

- Desired - analyze IP traffic from station A (on P2) to station B (on P3) and vice versa
- Implementation - add a PMAEntry and PMAExit filter to ports 2 and 3 with Protocol Type of 800(type IP in hex).

The reason for a PMAEntry and PMAExit filter is when A and B communicate there is communication both ways, i.e. IP packets are transmitted and received by P2.

3.11.5 Example #2: REMOTE Port Mirroring

Port 1 on ATX #1 is the diagnostic port where the analyzer resides.

Ports 2 on ATX #2 is the mirrored port.

Port 5 on ATX #1 has an ip address of 134.141.100.1.

Port 4 on ATX #2 has an ip address of 134.141.100.2 .

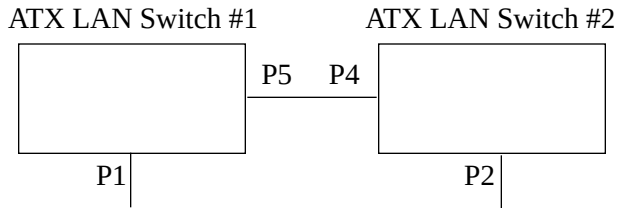
(P4 has to have an ip address assigned so ATX #2 will have an ip to ARP with.)

Config on
ATX #1

mirror
remote to 1
discard

or

mirror
remote to 1
truncate



Config on ATX #2

mirror remote 2 to 134.141.100.1

Mirror Filters with REMOTE Port Mirroring:

- Desired - to see packets from station A (on P2) only
- Implementation - add a PMEntry filter to port 2 on ATX #2 with station A's MAC address as the source address in the filter.

3.12 IPX ROUTING OVER SOURCE ROUTE COMMANDS

Command **ipxroute** is expanded with additional option **sr** to support IPX SR on token ring and FDDI ports. Option **sr** implies **on**. The explorer type and cache aging time can be configured using SNMP with the MIB variables. Refer to the ATX MIB Reference Guide Addendum (902021) for the specific MIB variables. The following is the LCM command format:

```
ipxroute [PORT-RANGE [{off | on | sr}]]
```

3.13 PING COMMANDS

The LCM command for ping is as follows:

```
ping [-rvsx] host_IP [data_size [count]]
```

-r = record route

-v = verbose

-s = send one packet per second continuously

-x = send packets continuously without delay

3.14 TRACEROUTE COMMANDS

The LCM command for trace route is as follows:

```
traceroute [-m max_ttl] [-q nqueries] [-w wait]  
host_IP [data_size]
```

3.15 EVENT LOGGING COMMANDS

The Event Log is established using the LCM. New LCM commands have been added in order to manage the event logging. There are 3 new LCM commands:

3.15.1 eventfilter

The LCM command format is:

```
eventfilter [clear | [overwrite |
stopwhenfull] [add|delete][allentries !
[filter_name[,filter_name]*] ]]
```

Examples:

eventfilter - prints out current eventfilter values

eventfilter arp - replaces current eventfilter with arp

eventfilter delete arp_request_timeout - deletes entries from current eventfilter

eventfilter allentries - turns on all entries in event filter

eventfilter clear - turns off event logging

eventfilter stopwhenfull arp - replace current eventfilter with arp, and stop keeping logging entries when the buffer gets full

eventfilter overwrite - keep current eventfilter value, overwrite buffer entries if necessary

The default event filter will be empty, meaning that no event logging entries will be kept. If the **eventfilter** command is issued without any options, the current **eventfilter** will be displayed. If the **eventfilter** command is issued without either an "add" or "delete" option, the entire **eventfilter** will be replaced. An **eventfilter** command issued with a "clear"

option will turn off event logging. The event logging entries will be kept in a circular buffer, and the logging entries will be overwritten if necessary. If the “stopwhenfull” option is given, the logging mechanism will cease entering logging entries into the event logging queue once it is full. By default, entries will be overwritten.

3.15.2 eventtrap

The LCM command format is:

eventtrap on|off

The eventtrap command will be used to determine whether event logging entries will trigger SNMP traps. By default, events generating SNMP traps will not be enabled.

3.15.3 eventdisplay

The LCM command format is:

eventdisplay [continuous]

The eventdisplay LCM command will get event logging entries that are currently in the event logging queue.

The eventdisplay command will output continuous event log entries, if specified, or the number of entries currently in the event logging queue. The continuous display of the event logging information to the console will be turned off by a Control-C.

3.16 CONFIGURING SOURCE ROUTE TRANSLATIONAL BRIDGING (RIF CACHING)

SRTB allows the ATX to strip and cache routing information for source route frames. Routing information (RIF) is used in source route networks to indicate the path a frame has taken through the network. This feature will enable the ATX to switch between source route only networks like Token Ring and transparent

networks like Ethernet and FDDI. RIF is not supported on Ethernet networks and is seldom used on FDDI networks. In order to merge source routed Token Ring networks with transparent Ethernet or FDDI networks the ATX must strip the RIF when communicating to Ethernet or FDDI and insert the RIF when communicating back to Token Ring. Source route networks contain the following features and parameters:

- SRTB is a global parameter and is enabled only on Token Ring ports with SRT bridging mode.
- The RIF database will support 8,192 entries.
- SRTB can be enabled based on protocols IP, IPX and Other(SNA, NetBIOS, etc...).
- All existing protocol translations(IP, IPX, SNA, NetBIOS and AppleTalk) will be supported when SRTB is enabled.
- The RIF caching aging timer is the same as the Spanning Tree timer, configurable and defaults to 300 seconds.
- The RIF cache entry is relearned based on a separate timer that is set to one half the Spanning Tree timer.
- A redundant/load sharing Source Route network is NOT supported when SRTB(RIF caching) is enabled. A redundant/load sharing SR network could have multiple paths to the Transparent network and cause the learning database to learn addresses on the incorrect ports. This could result in frames not getting forwarded and loss of communication. See Example #4.

3.16.1 Managing SRTB

SRTB [{ip | ipx | other |all}]{on | off}][ste | are]

IP - enables stripping and caching of RIF on IP frames

IPX - enables stripping and caching of RIF on IPX frames

Other - enables stripping and caching of RIF on AppleTalk, SNA and NetBIOS frames

All - enables stripping and caching of RIF on IP, IPX and OTHER (default when enabled)

On - enables SRTB globally; enabled per port when SRT is switching mode

Off - disables SRTB globally (default)

STE - enables the ATX to use a Spanning Tree Explorer frame when transmitting onto a Source Route network that it does not have a RIF entry for (default when enabled)

ARE - enables the ATX to use a All Route Explorer frame when transmitting onto a Source Route network that it does not have a RIF entry for.

3.16.2 SRTB Usage in the ATX

The table below describes the results of the RIF; it does not take into consideration frame translations. The translations will work the same as before. For this table, it is assumed that a transparent (TP) frame enters a TP configured port and a SR frame will enter a SR port. Either a TP or SR frame can enter a port configured for SRT.

Table 3-3 SRTB USAGE IN THE ATX

Entrance Port Config	Exit Port Config	Strip and Cache	Append RIF from Data Base	Forward as:
TP (Eth, FDDI, or TR)	SRT (TR only)	NO	YES	ARE OR STE (null RIF)
SR (TR only)	SRT (TR only)	NO	NO	SOURCE ROUTE

Table 3-3 SRTB USAGE IN THE ATX

Entrance Port Config	Exit Port Config	Strip and Cache	Append RIF from Data Base	Forward as:
SRT-TP FRAME (FDDI or TR)	SRT (TR only)	NO	YES	ARE OR STE (null RIF)
SRT-SR FRAME (FDDI or TR)	SRT (TR only)	NO	NO	SOURCE ROUTE
SRT-TP FRAME (TR only)	SRT (TR only)	NO	YES	SOURCE ROUTE
SRT-SR FRAME (TR only)	SRT (TR only)	NO	NO	SOURCE ROUTE
SRT (TR only)	TP (Eth, FDDI or TR)	YES	NO	TRANSPARENT
SRT (TR only)	SR (TR only)	NO	NO	SOURCE ROUTE
SRT (TR only)	SRT (FDDI or TR)	YES	NO	TRANSPARENT

TP- transparent SR- source route SRT- source route transparent (non token ring port) SRT / TR- source route transparent on a token ring port, i.e., SRTB enabled.

Notes: RIF caching can only be performed on a Token Ring port when it is set for SRT mode and is either the entrance or exit port. If there is a station directly attached to the ATX and it sends out a frame with a NULL RIF the ATX will not cache that station with RIF associated with it and treat it as a transparent station. A NULL RIF includes only two bytes with no bridge or ring numbers because it has not traversed any bridges. See example 2 on page 47. The ATX can only cache the RIF on IP, IPX, AppleTalk, SNA

and NetBIOS frames. All other protocols will NOT have their RIF cached. Support for other protocols will be in future releases.

Example #1:

Port 1 is configured for transparent

Port 2 is configured for source route transparent

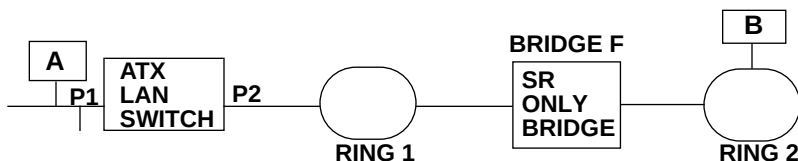
SRTB is enabled

LCM Commands:

bridge 1 transparent

bridge 2 SRT

SRTB all on ARE



Station A sends out a broadcast for station B. The ATX will see that station A resides on P1 and enter station A into the Bridge Address Table with no RIF associated. The ATX will then send an ARE (Null RIF) out P2. Station B receives the ARE and replies to station A. The ATX receives the reply on P2 and enters station B into the Bridge Address Table with the RIF of Ring 2, Bridge F and Ring 1.

The next time station A speaks to station B the frame will enter P1 on the ATX transparently and be sent out P2 with the RIF of Ring 1, Bridge F and Ring 2.

In previous releases, without the SRTB option enabled, the packet from Ethernet A to Token Ring B would have been forwarded out P2 as a transparent packet and never crossing the SR only bridge.

Example 2:

Port 1 is configured for Transparent

Port 2 is configured for Source Route Transparent

Port 3 is configured for Source Route

SRTB is enabled globally

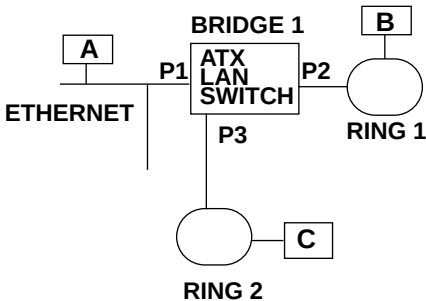
LCM Commands:

bridge 1 transparent

bridge 2 SRT

bridge 3 SR

SRTB all on ARE



Station A sends out a broadcast for station B. The ATX will see that station A resides on P1 and enter station A into the Bridge Address Table with no RIF associated. The ATX will then send an ARE (Null RIF) out P2. Station B receives the ARE and replies to station A. The ATX receives the reply on P2 and enters station B into the Bridge Address Table with no RIF associated. The reply from station B will not have any Ring or Bridge numbers so the ATX will know it is directly attached and the session between A and station B will be transparent.

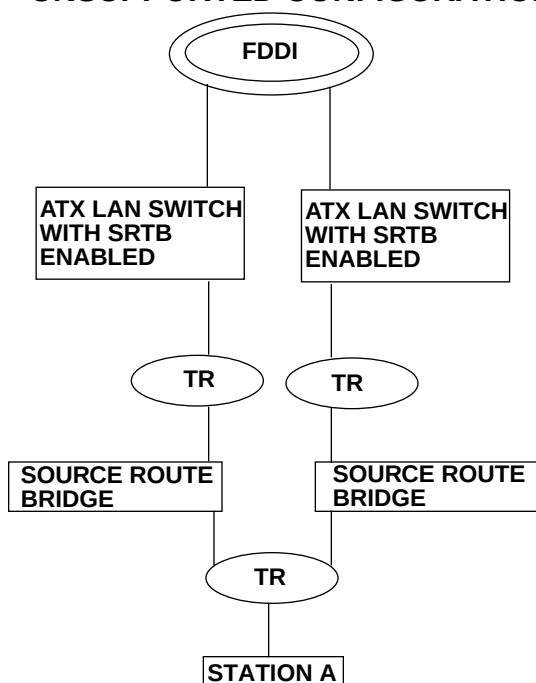
Scenario 2

Station C sends out a broadcast for station B. The frame from station C will have a Null RIF (2 bytes). Since the ATX's P3 is configured for SR, the ATX will add Ring 2, Bridge 1 to the frame and send it out P2. The conversation between station B and C will be source routed and the ATX will behave like a SR bridge. The conversation between station B and C will be source routed and the ATX will behave like a SR bridge.

Example #3: Unsupported Configuration

Problem

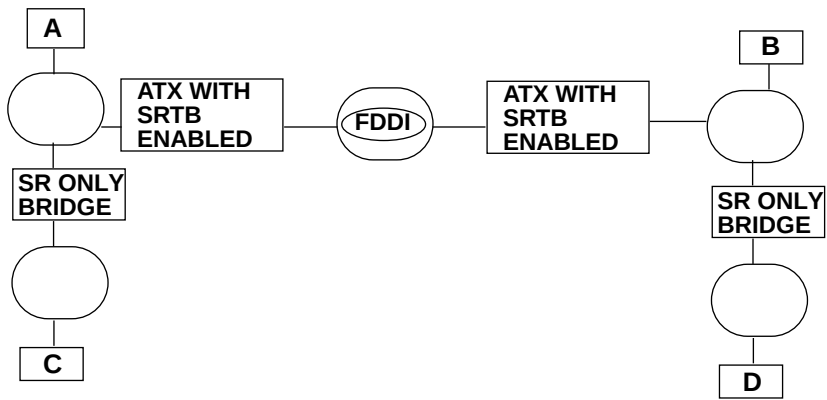
If Station A sends out an All Routes Explorer packet destined for Station B it will initially be learned by both ATXs on their Token Ring interfaces. After the ATX performs the Strip RIF portion it will forward the transparent frame to FDDI. This results in two transparent packets on the FDDI with the same Source MAC address (Station A). This causes each ATX to relearn Station A on its FDDI interface. The response from Station B to Station A will be seen by each ATX's FDDI interface and not be forwarded to the Token Ring because Station A was last learned on their FDDI interface.

UNSUPPORTED CONFIGURATION**Example #4: Maximum Transmit Unit**

The maximum frame size on FDDI is 4500 bytes, maximum on Ethernet is 1518 bytes and the maximum on Token Ring is 17800 bytes. As you can see when transmitting a TR frame over Ethernet or FDDI via a bridge, there could be a frame size conflict. The mechanism for Token Ring end stations to be notified of the maximum frame size a bridge will forward is in the Routing Control field. In this field the bridge can select one of seven frame sizes (516, 1470, 2052, 4472, 8144, 11407, 17800 according to the TI TMS380 guide) to notify the end station that this is the maximum frame size this bridge will forward.

For the ATX, this field is selectable via SNMP and defaults to 8144 for 16MB rings and 4472 for 4Mb rings. The scenarios below

describe a few configurations and possible solutions that address this problem.



Scenario 1: Local Stations

Problem:

Assume station A has already communicated and the ATX has learned it as a local transparent station. If station A has a MTU of anything greater than 4500 and wants to transfer a file to station B, it will not work.

Reason:

The ATX has no way of telling station A that it can not handle frames greater than 4500 because it is not passing traffic to station A with the Routing Control field. Station A will be able to connect to station B. However, if a file transfer is requested by A it will try to transfer that file at the largest frame size that A and B have negotiated. Assuming that B is configured the same as A, this file transfer will not happen because the maximum frame size on FDDI is 4500.

Solution:

The end station has to be configured for a MTU of less than the maximum frame size of the transport media (i.e., FDDI, Ethernet or Fast Ethernet).

Scenario 2: Stations across a Source Route only bridge

Problem:

Assume station C has already communicated and the ATX has learned the RIF associated with it. If station C has a MTU of anything larger than 4500 and wants to transfer a file to station D, it will not work.

Reason:

The ATX does have the Routing Control field to tell station C that it can't handle a 4500 frame size but the default for the ATX is 8144.

Solution:

Configure the **dot1dSrPortLargestFrame** OID with 4472, 2052 or 1470 depending on the MTU of the end station and the transport (i.e., FDDI, Ethernet or Fast Ethernet).

CHAPTER 4

MONITORING AND MANAGING THE ATX

Monitoring your ATX consists of collecting and analyzing statistics and status information. You can use LCM to gather some information, but you need to use an NMS as your primary tool. Managing your ATX consists of bringing modules on or offline, disabling or enabling ports, setting the community name for the ATX, and changing the console baud rate, all of which can be done using LCM.

4.1 MONITORING STATISTICS

The ATX collects statistics that can assist you to build a comprehensive profile of the traffic flow on each network, between networks, to and from each end-node within your network, and from outside your network. You can identify combinations of source network, destination network, source end-node, destination end-node, and protocol type for collection of traffic statistics. This is made possible by the ATX's multiple RISC processors.

Each ATX is capable of compiling statistics for all attached networks and 8,192 end-nodes. You can use this information, available through most SNMP-based NMSs, to analyze your network traffic flow and to make configuration changes as necessary. You can then head off potential bandwidth bottlenecks before they occur.

The end-node information can help you identify nodes that require high bandwidth and should be connected through a dedicated connection, rather than a shared, network connection. It can also help identify an end-node that is generating many multicast packets due to a malfunction.

For a more detailed analysis, you can have the NMS combine statistics for source network, destination network, source end-node, destination end-node, and protocol type.

Statistics that apply to the ATX as a whole are described here and the applicable MIB variable is provided. ATX statistics are divided into six general groups:

- General status and statistics
- IP status and statistics
- ICMP status and statistics
- UDP status and statistics
- SNMP status and statistics
- Spanning Tree status and statistics.

Note: *All statistics counters are cleared when the ATX is reset. Counters for individual ports are reset when the module is disabled and then re-enabled.*

Module statistics are generic to all modules and are included in this chapter. Port statistics vary depending on the type of module you are using. Port statistics are included in the individual module documentation.

The following are the statistics available for each network in the system:

- Number of packets generated by all end-nodes on each network divided into the type of destination address.
- Number of packets forwarded from each network for each type of destination address.
- Number of packets filtered divided into reason for filtering.
- Number of bytes generated by end-nodes on each network.
- Number of bytes forwarded or filtered from each network.
- Number of packets on each network that were sourced from outside that network divided by type of destination address.
- Number of bytes on each network that were sourced from outside that network.

- Number of packets that were sourced from outside a network that were not forwarded to the network.
- Number of packets with CRC errors on each network.

The following are the statistics collected by the ATX for each end-node:

- Number of seconds since the end-node last sent a packet on the network.
- Number of packets generated by the end-node.
- Number of bytes generated by the end-node.
- Number of packets generated by the end-node with destination outside of its network.
- Number of packets destined for the end-node that were sourced outside of its network.
- Number of bytes destined for the end-node that were sourced outside of its network.
- Number of multicast packets generated by the end-node.

The following are the statistics collected by the ATX for filters:

- Number of packets filtered.
- Address of the last source of the packet that satisfied the defined criterion.

You can also collect statistics that profile the traffic by protocol type or between pairs of end-nodes. For example, you could profile the traffic between pairs of end-nodes, on a particular network by protocol type, by protocol type for a particular end-node, between networks by protocol type, and between end-nodes by protocol type.

4.1.1 General Status and Statistics

The following statistics profile the general status of the ATX. (The MIB variable that collects the statistics is provided in square brackets.)

- The number of centiseconds (hundredth of a second) since the ATX was last reset. [sysUpTime]
- What the ATX is being used for: bridging, IP Routing, or Bridging and IP Routing. [sysServices]
- The physical location of the ATX. [sysLocation]
- The name and address of the contact person for the ATX. [sysContact]
- The name of the ATX. [sysName]
- Number of packets not received due to internal buffer congestion. [ppeTxCongests]
- The number of ARP entries for all interfaces. [ppeArpEntries]
- The number of statically defined ARP entries for all interfaces. [ppeArpStatics]
- The number of times an ARP entry could not be learned due to insufficient address table space. [ppeArpOverflows]
- The number of IP Routing Database entries. [ppeIpEntries]
- The number of statically defined IP Routing Database entries. [ppeIpStatics]

4.1.2 IP Status and Statistics

The following statistics relate specifically to IP routing. (The MIB variable that collects the statistics is provided in square brackets.)

- The number of IP routes lost. [ppeRipRoutedDiscards]

- The total number of IP packets received from all ports (including the UART). [ipInReceives]
- The number of packets received that were discarded by IP due to errors in the IP header. [ipInHdrErrors]
- The number of packets received that were discarded by IP due to an invalid (or non-routable) destination IP address in the IP header. [ipInAddrErrors]
- The number of packets received that were routed by IP towards a final IP destination. [ipInForwDatagrams]
- The number of packets received that were addressed to this ATX's IP, but were discarded because of unknown or unsupported protocol. [ipInUnknownProtos]
- The number of packets that were received without error, but were not processed (due to insufficient resources, for example). [ipInDiscards]
- The total number of input packets successfully delivered to the IP user-protocol layers. [ipInDelivers]
- The total number of IP output packets generated by this ATX. This count does not include any received packets forwarded by this ATX. [ipOutRequests]
- The total number of output packets which were discarded (due to lack of resources, for example). This counter includes packets which would be included in ipForwDatagrams if any such packets were discarded. [ipOutDiscards]
- The number of packets which were discarded because no route could be found to transmit them to their destination. This counter includes any packets counted in ipForwDatagrams which meet this "no-route" criteria. [ipOutNoRoutes]
- The maximum time, in seconds, that received fragments are held while they are awaiting reassembly within this ATX. [ipReasmTimeout]

- The number of IP fragments received which needed to be reassembled within this ATX. [ipReasmReqds]
- The number of IP datagrams which were successfully re-assembled. [ipReasmOKs]
- The number of failures (for whatever reason: timed-out, errors, etc.) detected by the IP re-assembly algorithm. [ipReasmFails]
- The number of IP datagrams that have been successfully fragmented within this ATX. [ipFragOKs]
- The number of IP datagrams that have been discarded because they needed to be fragmented but could not be (their “Don't Fragment” flag was set). [ipFragFails]
- The number of IP datagram fragments that have been generated by this ATX. [ipFragCreates]
- The number of valid routing entries which were discarded. [ipRoutingDiscards]

4.1.3 ICMP Status and Statistics

The following statistics relate specifically to ICMP routing. (The MIB variable that collects the statistics is provided in square brackets.)

- The total number of ICMP messages which were received by this ATX. This also includes all messages received with errors. [icmpInMsgs]
- The number of ICMP messages which were received with errors (bad checksums, bad length, etc.). [icmpInErrors]
- The number of ICMP Destination Unreachable messages received. [icmpInDestUnreachs]
- The number of ICMP Time Exceeded messages received. [icmpInTimeExcds]

- The number of ICMP Parameter Problem messages received.
[icmpInParmProbs]
- The number of ICMP Source Quench messages received.
[icmpInSrcQuenchs]
- The number of ICMP Redirect messages received.
[icmpInRedirects]
- The number of ICMP Echo (request) messages received.
[icmpInEchos]
- The number of ICMP Echo Reply messages received
[icmpInEchoReps]
- The number of ICMP Time-stamp (request) messages received.
[icmpInTimestamps]
- The number of ICMP Time-stamp Reply messages received.
[icmpInTimestampsReps]
- The number of ICMP Address Mask Request messages received.
[icmpInAddrMasks]
- The number of ICMP Address Mask Reply messages received.
[icmpInAddrMaskReps]
- The total number of ICMP messages which were created by this ATX. This includes all messages counted by icmpOutErrors.
[icmpOutMsgs]
- The number of ICMP messages which this ATX did not send due to problems discovered entirely within the ICMP subsystem (such as lack of buffers). [icmpOutErrors]
- The number of ICMP Destination Unreachable messages sent.
[icmpOutDestUnreachs]
- The number of ICMP Time Exceeded messages sent.
[icmpOutTimeExcds]

- The number of ICMP Parameter Problem messages sent.
[icmpOutParmProbs]
- The number of ICMP Source Quench messages sent.
[icmpOutSrcQuenchs]
- The number of ICMP Redirect messages sent. [icmpOutRedirects]
- The number of ICMP Echo (request) messages sent.
[icmpOutEchos]
- The number of ICMP Echo Reply messages sent.
[icmpOutEchoReps]
- The number of ICMP Time-stamp (request) messages sent.
[icmpOutTimestamps]
- The number of ICMP Time-stamp Reply messages sent.
[icmpOutTimestampReps]
- The number of ICMP Address Mask Request messages sent.
[icmpOutAddrMasks]
- The number of ICMP Address Mask Reply messages sent.
[icmpOutAddrMaskReps]

4.1.4 UDP Status and Statistics

The following statistics relate specifically to UDP. (The MIB variable that collects the statistics is provided in square brackets.)

- The total number of UDP datagrams delivered to UDP users.
[udpInDatagrams]
- The total number of received UDP datagrams for which there was no application at the destination port. [udpNoPorts]
- The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port. This number is always zero, since the ATX handles resource limitations by discarding datagrams at the IP

level; all datagrams forwarded to UDP are always forwarded to the ATX's local management agent. [udpInErrors]

- The total number of UDP datagrams sent from this ATX. [udpOutDatagrams]

4.1.5 SNMP Status and Statistics

The following statistics relate specifically to SNMP. (The MIB variable that collects the statistics is provided in square brackets.)

- The number of SNMP PDUs received by the ATX. [snmpInPkts]
- The number of SNMP PDUs created by the ATX and passed to the PPE. [snmpOutPkts]
- The number of SNMP PDUs received by the ATX which had an unsupported SNMP version. [snmpInBadVersions]
- The number of SNMP PDUs received by the ATX which had an unrecognized SNMP community name. [snmpInBadCommunityNames]
- The number of SNMP PDUs received by the ATX which had an authentication failure. [snmpInBadCommunityUses]
- The number of SNMP PDUs received by the ATX which had an ASN.1 parsing error while being decoded by the ATX. [snmpInASNParseErrs]
- The total number of MIB objects which have been successfully retrieved by the ATX as a result of SNMP get request or getNext PDUs. [snmpInTotalReqVars]
- The total number of MIB objects which have been successfully altered by the ATX as a result of SNMP SetRequest PDUs. [snmpInTotalSetVars]

- The total number of SNMP GetRequest PDUs received by the ATX, which have been processed with no errors. [snmpInGetRequests]
- The total number of SNMP GetNext PDUs received by the ATX, which have been processed with no errors. [snmpInGetNexts]
- The total number of SNMP SetRequest PDUs received by the ATX, which have been processed with no errors. [snmpInSetRequests]
- The total number of SNMP PDUs created by the ATX, with a value of tooBig in the PDU's ErrorStatus. [snmpOutTooBigs]
- The total number of SNMP PDUs created by the ATX, with a value of noSuchName in the PDU's ErrorStatus. [snmpOutNoSuchNames]
- The total number of SNMP PDUs created by the ATX, with a value of badValue in the PDU's ErrorStatus. [snmpOutBadValues]
- The total number of SNMP PDUs created by the ATX, with a value of genErr in the PDU's ErrorStatus. [snmpOutGenErrs]
- The total number of SNMP GetResponse PDUs created by the ATX. [snmpOutGetResponses]
- The total number of SNMP Trap PDUs created by the ATX. [snmpOutTraps]

4.1.6 Spanning Tree Status and Statistics

The following statistics relate specifically to SNMP. (The MIB variable that collects the statistics is provided in square brackets.)

- The number of spanning tree topology changes which have occurred, since the ATX was last booted. [stTopChangeCount]

- Whether a topology change is currently in progress. [stTopChange]
- If a topology change is in progress then this is the time since the topology change was initiated. If a topology change is not in progress then this is the time since a topology change was finished. [stTopChangeTime]

4.2 MODULE STATUS AND STATISTICS

The status and statistics described in this section are applicable to all Input/Output Modules.

- Whether the module's temperature is too hot. [hwTempOK]
- Results of diagnostics, when diagnostics were last performed on the module (usually power-up). Possible values: diagnostics failed, diagnostics still running, diagnostics passed. [hwDiagStatus]
- Status code for the diagnostics that was last run on the module. [hwDiagCode]
- Whether the module is currently executing its operational software. [hwInuse]
- The manufacturing information for this module which includes serial number, hardware revision level, and serial number. [hwManufData]

4.2.1 End-node Status and Statistics

The ATX keeps statistics on end-nodes on attached networks, allowing you to monitor the traffic flows from those networks. Status and statistics on each end-node, recorded in the Bridge Address Table, follows:

- The port through which this station is connected to the ATX (only valid for dynamically learned addresses and unique addresses for ATX's agent software).

- The time, in centiseconds, since a packet was last received from the station.
- The number of packets received from the station which were forwarded.
- The number of packets transmitted to the station.

You can configure the ATX to collect extended statistics by using an SNMP Manager to set the MIB variable `ppeExtendStats` to one. The ATX is shipped with no extended statistics collection as the default. For each station, the following additional statistics are collected if you configured the ATX to collect extended statistics.

- The number of characters in the packets received from the station.
- The number of multicast packets received from the station.
- The number of characters transmitted to the station.

4.2.2 Traffic Analysis Statistics

You can configure the ATX to collect statistics on traffic between two end-nodes, in ways described below. Refer to Chapter 5, **Filters** for instructions on setting up filters.

- Number of packets sent from Station A to Station B.

Configure pseudo source-port filter with Station A's address as source address match and Station B's address as destination address match. [`filterPktCnts`]

- Number of IP packets sent from Station A to Station B.

Configure pseudo source-filter with Station A's address as source address match and Stations B's address as destination address match and Frame Type set to IP. [`filterPktCnts`]

- Number of packets sent from Station A to Segment B.

Configure pseudo destination filter on port B with Station A's

address as source address match. [filterPktCnts]

- Number of packets sent from Segment A to Station B.

Configure pseudo source filter on port A with Station B's address as destination address match. [filterPktCnts]

4.3 MONITORING STATUS

You can monitor the ATX with LCM, to see its status at a glance. The LCM commands that allow the monitoring the status of the ATX are described in the sections that follow.

4.3.1 Displaying Status

The status command displays the status of the entire ATX, and automatically pages through the status of all of the ports, pausing at each screen of information.

Note: Also use the status command to display status for individual ports. The information displayed from the **status <port number>** command varies from module to module and is therefore described in the individual module manuals.

Type: **status**

LCM displays the following type of information.

Current Number of Learned Addresses: 133

Number of Defined Filters: 4

PPE CPU utilization is light

Module	Type	DiagStatus	InUse	TempOK	Ports
1	PPE	Passed	True	Normal	1
2	FDDI-IOM	Passed	True	Normal	2
3	CSMA-IOM	Passed	True	Normal	3
4	CSMA-IOM	Passed	True	Normal	4
5	CSMA-IOM	Passed	True	Normal	5-8
6	TR-IOM	Passed	True	Normal	7-9
7	Turbo	Passed	True	Normal	

Type: <CR> to display port 2 status...

If you don't want to view the status of each port, use the Ctrl-C keys to return to the LCM prompt.

The status of the entire unit includes the number of learned addresses and the number of defined filters, plus the following information for each of the ATX's modules:

- **Type** – lists the module types; if there is no module present, the type will be listed as Vacant. Module types include:
 - **PPE** – the Packet Processing Engine (module 1)
 - **FDDI** – a Fiber Distributed Data Interface module
 - **TR** – a Token Ring module
 - **CSMA** – an Ethernet module, it could be a single port or four port module
 - **HSSI** – a High Speed Serial Interface module
 - **Turbo** – the turbo processor (module 7)
 - **DiagStatus** – whether the module passed or failed its power-up diagnostics. If the module failed, then a hexadecimal error code (hwDiagCode) is also displayed.
- **InUse** – whether the module is currently being used. A module that has passed its power up diagnostics is displayed as **true**, unless one of the following conditions is present:
 - The module was disabled using the **offline** command, or a remote NMS has disabled that module by setting the **hwUseMod MIB** variable.
 - The module has malfunctioned since its power-up diagnostics were executed.
 - The module was just physically inserted without following the recommended procedures.
 - The module was physically inserted earlier, but was not activated by the ATX at the time (due to the power being off, or the procedures not being followed correctly), and you are now turning on power to the unit.

- **TempOk** – indicates whether the module is overheating. Normal is displayed when the module temperature is within range. Too-Hot is shown for abnormal temperature status.
- **Ports** – lists the port numbers of the ports on the module.

4.3.2 Displaying MAC Addresses

The `addresses display any` command displays all MAC addresses. The display includes the MAC address, type of address, port number, age (in seconds since a packet was last received from that address), number of packets received from that address, and number of packets forwarded to that address. The type of address may be Learned, Port (for the MAC address assigned to a port), Static (for a Spanning Tree static address that was added by an NMS, or Other (for none of the above). The display automatically pauses with each screen of information.

Addresses are displayed in random order; for example, address 02:00:00:00:00:00 may appear after address 04:00:00:00:00:00. Since the Bridge Address Table changes dynamically, a learned address may be displayed twice (this could occur if an address is aged out after being displayed and then relearned before the end of the display).

The age will be the most recent of the following:

- Time since a packet was last received from that address.
- Time since the ATX last created a packet with that source address.
- Time since the ATX created that address.

To display MAC addresses:

Type: **addresses display any**

LCM responds with a list of all MAC addresses, their associated ports, the type, age, and number of frames from and to that address.

Address	Type	Port	Age(secs)	Frames-from	Frames-to
08:00:20:02:3a:44	Learned	3	26	1	0
00:40:27:03:b7:21	Static	5	5	17110	195
00:80:20:a2:3b:0a	Other	4	1	1423	121

Enter <CR> to continue, Ctrl-C to exit:

To display a specific address:

Type: **addresses display <MAC address>**

For example, if you typed, **addresses display 02:04:06:03:2a:43**, LCM would display the following information:

Address	Type	Port	Age(secs)	Frames-from	Frames-to
02:04:06:03:2a:43	Learned	5	21	1181	73

You can display a range of addresses by using a net mask:

Type: **addresses display <MAC address> <net mask>**

For example, to see all addresses that begin with 02:04:06 you would enter:

```
addresses display 02:04:06:00:00:00  
ff:ff:ff:00:00:00
```

LCM would display:

Address	Type	Port	Age(secs)	Frames-from	Frames-to
02:04:06:03:2a:43	Learned	5	21	1181	73
02:04:06:00:2a:67	Learned	4	1	3421	0
02:04:06:a3:70:2b	Learned	5	0	15339	235

Enter <CR> to continue, Ctrl-C to exit:

To display addresses in Token Ring or FDDI native bit order:

Type: **addresses display big**

For example, if you typed, **addresses display 02:04:06:03:2a:43**, LCM would display the following information:

Address	Type	Port	Age(secs)	Frames-from	Frames-to
10 00 90 c1 d1 1d	Learned	6	0	1036886	8624
10 00 04 20 c9 39	Learned	6	0	63995	4432

4.3.3 Displaying Manufacturing Information

The `ident` command identifies ATX manufacturing information, including the version of software that has been saved in flash memory, the part number, revision number, and serial numbers of all of the modules. It also displays the length of time since the ATX was last rebooted.

To display the manufacturing information:

Type: **ident**

LCM displays the following type of information:

```
Software Currently Running: Version 2.3 12-JAN-94
Next Bootstrap: Version 2.3 12-JAN-94
```

```
System Up Time: 4 days, 11:03:06
```

Slot	Module Type	Part No.	Rev	Serial No.
1	Packet Processing Engine	512-0050	003	0002e4000c61
2	FDDI DAS	512-0014	009	0002E40001F7
3	Ethernet	512-0015	014	0002E400036a
4	HSSI/T1	512-0068	01A	0002E400054a
5	Quad Ethernet AUI	512-0044	002	0002E40002b4
6	Token Ring	512-0046	001	0002E400067a

4.4 MANAGING YOUR ATX

This section describes how you can manage your ATX by enabling and disabling specific ports, turning modules on or offline, changing the baud rate of your terminal connection or adding or changing a community name.

4.4.1 Disabling a Port

There may be times when you need to disable a specific port. Disabling a port effectively stops all of the bridging and IP routing functions for that port. Ports that have been disabled won't be able to accept SNMP packets, and therefore can't communicate with an NMS.

To disable a port or port range:

Type: **disable <port range>**

For example, **disable 7-9** would disable ports 7, 8, and 9. LCM responds:

Port 7: Bridging/Routing functions DISABLED!

Port 8: Bridging/Routing functions DISABLED!

Port 9: Bridging/Routing functions DISABLED!

***Note:** Once a port is disabled, it will be disabled until you enable it again. Resetting the ATX won't enable a port that has been disabled.*

***Caution:** If you disable the port through which you are connected to the ATX, you will not be able to communicate with the ATX. To find the port number through which you are connected, use the **addresses display** command with the MAC address of your device.*

4.4.2 Enabling a Port

When you enable a port that has been disabled, whatever bridging and routing functions you had configured for that port would then be enabled. You may want to enable a port even if no bridging or routing functions are configured to allow the port to communicate with an NMS.

To enable a port or port range:

Type: **enable <port range>**

For example, **enable 7-9** would enable ports 7, 8, and 9. LCM responds:

Enabling bridging/routing functions for port 7
Enabling bridging/routing functions for port 8
Enabling bridging/routing functions for port 9

You can use the `bridge` and `iproute` commands to see what functions are configured or to change the configuration.

4.4.3 Taking a Module Offline

If you need to take a module offline, you can use LCM, or you can use the reset button on the module itself. (On newer modules the button is labelled **Offline**, on older modules it is labeled **Reset**.) Module 1, the Packet Processing Engine, and Module 7, the Turbo processor, cannot be stopped. Modules 2 through 6 are the I/O modules and they can be stopped.

To take a module offline:

Type: **offline** <module number>

For example, **offline 6** would take module 6 offline.

***Caution:** If you issue an offline command for the module through which you are connected to the ATX, you will not be able to communicate with the ATX.*

4.4.4 Bringing a Module Online

To bring a disabled module back online, use the `online` command. Only I/O modules may be restarted.

To bring a module back online:

Type: **online** <module number>

For example, **online 6** would bring module 6 online.

4.4.5 Setting The Baud Rate

You can set the baud rate for your LCM console connection. The options for baud rate include:

- 1200
- 2400
- 4800
- 9600
- 19200

***Note:** Make sure that the baud rate you set matches the baud rate setting for the terminal you are using.*

The default rate is 9600. To change the setting:

Type: **baud <baud rate>**

For example, **baud 4800** would set the baud rate to 4800.

LCM responds:

Baud rate is 4800.

4.4.6 Displaying The Baud Rate

To display the current setting:

Type: **baud**

LCM responds:

Usage: **baud [1200|2400|4800|9600|19200]**

Baud rate is 4800.

4.4.7 Assigning a Community Name

A community name is similar to a password. You use the same steps to assign new or change existing community names. This sets

the MIB variable `configAnyPass`; you must then enter the community name to perform any gets or sets. What you type is not echoed to the screen, so you won't see what you are typing.

To assign a community name

1. Type: **community**
2. Enter the old community name.
3. If one hasn't been assigned, you don't need to enter anything. LCM prompts you for the new community name.
4. Enter the new community name.
5. LCM prompts you to verify the new community name by retyping it.
6. Retype the new community name.

CHAPTER 5

FILTERS

One of the most significant features of the ATX is its powerful user-configurable filtering capabilities. Flexible filtering is useful for implementing security measures and improving network performance. For some applications, filtering capabilities may be so important that they are the primary reason for using a bridge.

A filter is an instruction to the ATX to screen packets based on the criteria you select. All bridges by nature filter packets; they discard local traffic. Local traffic is defined as packets that are destined for the same network from which they came.

In addition to the basic bridge function of filtering local traffic, the ATX allows you to implement two types of filters that are useful for managing and administering networks:

- Address table filters, which use the Bridge Address Table to screen local traffic.
- Combination port filters, which apply filters to or from a specific port segment.

Filters should be used judiciously, because they may degrade network performance. (See **Filtering and Performance Considerations** below.)

The ATX can be configured to selectively filter network traffic based on source or destination address, entry or exit port, type of packet or a custom mask applied anywhere in the data packet. An entry port is defined as a pre-processing filter; the filter condition is satisfied first and then a bridging decision is made. An exit filter is defined as a post-processing filter; the ATX makes a bridging decision and then acts on the filter. Based on selection parameters you define, the ATX identifies data packets that are to be filtered and discards them.

The following sections describe the ATX's enhanced filtering capabilities. "Adding a filter" describes how to set up a filter.

5.1 FILTERING AND PERFORMANCE CONSIDERATIONS

When filters are implemented, the ATX must process packets to determine if they should be filtered. The processing that takes place on filters can therefore exact a toll on ATX throughput (or forwarding) performance. Typically, if you are using address table filters or a small number of combination port filters, they will have little effect on performance. However, a large number of combination port filters will reduce the maximum possible forwarding rate. For this reason, filters that are no longer needed should be removed.

5.2 USING FILTERS FOR SECURITY PURPOSES

The various types of security restrictions that can be implemented using ATX filters are summarized below:

- **Restrict access to a physical segment** – A filter can be configured to prevent any traffic from being forwarded to a specific network segment. If, for example, a filter is configured to block all traffic to the port that connects LAN A to the ATX, all access to LAN A will be restricted.
- **Restrict access to a host device** – Filters can be configured to restrict access to a host device in a variety of ways. For example, filters could be configured to impose either of the following conditions:
 - Only users assigned security level A can use host computer X.
 - Users assigned security levels C and D cannot use computer Y.
- **Restrict end-nodes** – Filters may also be used to restrict individual workstations from accessing other network devices. For example, filters could be configured to impose either of the following conditions:
 - User B1 can only access workstations C2 and C3.
 - User B1 cannot access workstation C12.

Detailed examples of filter applications are presented later in this chapter. (See **Filtering Application Examples**.)

5.3 USING FILTERS TO IMPROVE PERFORMANCE

In many applications, ATX filters can be used to enhance network performance by preventing certain types of traffic which may degrade performance. A filter that defines logical barriers to protect a network segment or segments from conditions that may degrade network performance is referred to as a firewall filter.

Firewalls define logical barriers to protect a network segment or segments from conditions that may degrade network performance. Examples of degrading conditions that may be controlled by firewall filters include unnecessary traffic, broadcast storms, and conflicting applications that occur within a particular segment. Firewall filters can also be used to help implement fault isolation, error recovery, and security measures.

A firewall filter may be an address table filter or a combination port filter. Three examples of firewall applications are described below. Firewall filters can be configured to:

- Allow server traffic only to be forwarded from LAN A to LANs B and C. (Other traffic would not be forwarded.)
- Prevent a specific type of traffic from being forwarded to a specific network segment. (For example, it might be desirable to block DECnet broadcast traffic from a LAN that includes no devices that use DECnet data packets. This would be an example of blocking unnecessary traffic.)
- Prevent multicast packets from being forwarded to a specific network segment (localized broadcast storm prevention).

An example of a firewall filter is given in the section, **Filtering Application Examples**.

Note: *The ATX multicast storm protection feature may be thought of as a firewall feature, in that it performs a protective blocking function,*

*but it is not a filter. Multicast storm protection is described in Chapter 3, **Configuring Multicast Storm Protection**.*

5.4 ADDRESS TABLE FILTERS

The simplest type of filters are address table filters. These filters use the Bridge Address Table to screen local traffic. To make highly efficient address filtering possible, the ATX address table includes filter flags. By setting these flags, a system operator can selectively filter:

- Traffic to and /or from any station (MAC layer address)
- Multicast traffic from any station (MAC layer address)

The capacity of the ATX address table is 8,192 entries expandable to 16,384. Of these, 200 can be static (manually entered) entries and a small number of special reserved addresses; the rest are dynamically learned addresses. Table 5-1 shows what one dynamically learned entry in the Bridge Address Table would look like.

Table 5-1. Representation Of Address Table Entry

MAC address	Port (segment)	Age	Source filter	Multicast source filter
00:01:02:03:04:05	3	26	OFF	OFF

The key entry in the address table is the MAC address which is either the Ethernet, Token Ring, or FDDI address. The port entry indicates the physical port or segment associated with the address. The segment port number is automatically learned for dynamic addresses, but may be manually entered if a static address is desired. The age entry indicates when a frame from the device was last received by the ATX. The source filter and multicast source filter entries are flags used solely for filtering; they instruct the ATX

to filter (ON) or not filter (OFF) packets from the specified address.

With the address table entry shown in Table 5-1, you could use any of the three types of address table filtering which are described in the sections that follow:

- Destination address filter
- Source address filter
- Source address multicast filter

5.4.1 Destination Address Filter

A destination address filter may be used to discard all traffic destined to a specific MAC address. This type of filter is configured by setting a static address entry for the MAC address and specifying {null} as the port assignment as shown in Table 5-2. The port assigned by the static entry will take precedence over the port learned by the ATX's learning algorithm.

Table 5-2. Representation Of A Destination Address Filter

MAC address	Port (segment)	Age	Source filter	Multicast source filter
00:01:02:03:04:05	{null}	26	OFF	OFF

The effect of this filter is that packets to the specified address will not be forwarded to any port.

5.4.2 Source Address Filter

The source address filtering capability uses the source filter flag, which is a component of each entry in the address table. When the flag is set to ON, all packets originating from the designated MAC address will be filtered.

An example of a source address filter is shown in Table 5-3. For illustration purposes, this example uses the same format as the address table entry shown previously. The actual format used for configuring filters depends on the NMS you use.

Table 5-3. Representation Of A Source Address Filter

MAC address	Port (segment)	Age	Source filter	Multicast source filter
00:01:02:03:04:05	2	26	ON	OFF

Because the source filter flag is set to ON, the effect of this filter will be to discard any traffic from the specified MAC address.

5.4.3 Combination Address Filters

The combination address filter capability uses source address filtering along with destination address filtering. In the example shown in Table 5-4, the combination address filter is discarding packets to the specified address (26:14:11:2:3:3) as well as discarding any traffic from the specified MAC address.

Table 5-4. Representation Of A Combination Address Filter

MAC address	Port (segment)	Age	Source filter	Multicast source filter
26:14:11:2:3:3	{null}	14	ON	OFF

5.4.4 Source Address Multicast Filter

The source address multicast filtering capability uses the multicast source filter flag in the address table as shown in Table 5-5. When this flag is set to ON, all multicast packets originating from the

designated MAC address will be filtered. Multicast packets are those destined for more than one address (using a multicast destination address). This is useful for preventing broadcast traffic from a particular station.

Table 5-5. Representation Of A Source Address Multicast Filter

MAC address	Port (segment)	Age	Source filter	Multicast source filter
00:01:02:03:04:05	2	12	OFF	ON

Because the multicast source filter flag is set to ON, this filter will discard any multicast packets from the specified MAC address.

5.5 COMBINATION PORT FILTERS

In contrast to address table filters, which apply to traffic to or from a particular MAC address, combination port filters apply to traffic to or from a specific port segment. They also provide greater filtering flexibility.

Combination port filters include multiple filtering conditions. This makes it possible to configure very specific filters. For example, a combination port filter could be structured to filter all AppleTalk packets from port segment 2 whose destination address is xyz. In this example, three filtering conditions are specified. The filter could be logically represented as:

- Filter packets if they are from port 2
- And if they are AppleTalk packets
- And if the destination address is xyz

The various types of filtering conditions that may be specified are referred to as fields. Two examples of fields are entry or exit port and Ethernet protocol type. The available fields for combination

port filters are described in the next section.

The ATX allows you to implement up to 100 combination port filters (total, for all connected ports). Combination port filters may be logically linked to one another as described previously in the AppleTalk example.

Each combination port filter generates statistics when invoked, and thresholds can be set to trigger alarms to the NMS.

5.5.1 Configurable Fields

Each combination port filter may contain entries for all seven of the configurable fields (with the exception that the entry port and port/group match fields may only be included in an exit port filter). If no value is specified for a particular field, that field will not be used for that filter. The port field must always be specified, since it identifies which traffic flow the ATX is to observe for filtering. If only the port is specified, the filter will screen no packets to or from that port (depending on whether it is an entry (source port filter) or an exit (destination port filter), because none of the filtering criteria fields were set.

The source address, destination address, protocol type, custom mask, and source port fields may be defined as **True**, **False**, or **Not applicable**. These definitions specify how the information in each field is to be applied:

- **True** – Means all traffic that matches the field will be filtered.
- **False** – Means all traffic that does not match the field selection will be filtered (inverse filter).
- **Not applicable**—Means that when the filter is invoked, the ATX will not check for this field. (This is the default selection for each field.)

In addition to the configurable fields, there are also two options that you can use with all filters:

- **Pseudo** – allows you to create a pseudo filter to monitor traffic patterns without discarding packets.
- **And/Or** – allows you to combine multiple port filters using the and /or operators to create boolean filter expressions.

These options are discussed in detail in the section “Combination port filter options”.

When you are adding filters, you are prompted for all of the possible field value options. The prompt usually includes a default value in brackets that will be used if you don’t specify a value. Configurable fields are described in the sections that follow.

Type

Either **Entry** (apply the filter to all packets received on the port) or **Exit** (apply the filter before transmitting the packet from the port). **Entry** is the default.

***Note:** Use PMEntry and PMExit to filter when using the port mirroring option.*

Source Range

Either **NA** (not applicable), **True** (filter the packet if the source MAC address is within the range), or **False** (filter the packet if the source MAC address is outside of the range). **NA** is the default.

Source Range Start

The starting MAC address for the source range of MAC addresses.

Source Range End

Ending MAC address for the source range of MAC addresses. If you are filtering on a single address, no entry is required.

Source Range Mask

MAC address mask to apply to the range of source MAC addresses. **ff:ff:ff:ff:ff:ff** is the default.

Destination Range

Either **NA** (not applicable), **True** (filter the packet if the destination MAC address is within the range), or **False** (filter the packet if the destination MAC address is outside of the range). **NA** is the default.

Destination Range Start

Starting MAC address for the destination range of MAC addresses.

Destination Range End

Ending MAC address, for the destination range of MAC addresses.

Destination Range Mask

MAC address mask to apply to the range of destination MAC addresses. **ff:ff:ff:ff:ff:ff** is the default.

Port/Group Match

Either **NA** (not applicable), **True** (filter the packet if the receiving port or group number matches), or **False** (filter the packet if the receiving port or group number does not match). This is valid only if the filter type is Exit. **NA** is the default.

Port/Group

Decimal value for the number of the port or group through which the packet entered the ATX. This is valid only if the filter type is

Exit. **NA** is the default.

Note: You can assign a filter to a group by entering a group number rather than a port number. You can assign a group number to specified ports using an NMS. Port group numbers start at 22.

Protocol Match

Either **NA** (not applicable), **True** (filter the packet if the protocol type matches), or **False** (filter the packet if the protocol type does not match). **NA** is the default.

Protocol Type

LLC, for all LLC frames; **Ethernet**, for all Ethernet-2 frames; or a hex value for the Ethernet Protocol Type field. All of the Ethernet hex values are listed in RFC 1060. Some common Ethernet protocol hex values include:

- 0800 – IP
- 0806 – ARP
- 6003 – DECnet Phase IV
- 809B – AppleTalk
- 80F3 – AppleTalk AARP

Field Match

Either **NA** (not applicable), **True** (filter the packet if the masked value matches), or **False** (filter the packet if the masked value does not match). This option allows you to examine a portion of a packet to set up customized filters to match conditions you specify. **NA** is the default.

Field Origin

Either **IP**, **MAC**, or **SR** (see Field Offset below). The origin is the field from which the offset count starts. **IP** is the default.

Field Offset

The decimal offset of the portion of the packet (as stored in canonical format) to be examined. If the origin is IP, then the offset is relative to the end of the IP Header (an offset of zero indicates the portion immediately following the end of the IP Header). If the origin is MAC, then the offset is relative to the beginning of the canonical format MAC addresses (an offset of zero indicates the start of the destination MAC address). If the origin is SR, then the offset is relative to the end of the MAC header, including the SR header if present. **Zero** is the default.

***Note:** The canonical format is described in Appendix B, **Packet Translation Procedure**, of this manual.*

Field Value

The hexadecimal value of the eight octets beginning at the origin and offset by the value specified above. The octets must be separated by spaces. This is the value that the filter is using when it does a comparison for a match, for example a MAC address.

Field Mask

An eight octet mask applied to the packet's eight octets before comparing them to the Field Value specified above. The mask octets must be separated by spaces. This is a mask of the specified Field Value.

Threshold Time

Number of seconds for which the threshold packet count applies.

Values greater than 3600 (one hour) are not valid; a value of zero indicates that no alarms should be generated. **Zero** is the default.

Threshold

Number of occurrences allowed within the specified threshold time; occurrences above this number cause an alarm to be generated. (The ATX's `configAlarmDynamic MIB` parameter must be set.) **Zero** is the default.

Filter Index

Filter number for this filter. For example, a value of one indicates that this is the first filter in the filter table. If you use the default index of 1, any other filters you have defined will be renumbered starting with 2. Although filters are assigned to a port, filter indexes are not; they are assigned sequentially to all filters for all ports. **One** is the default.

Combination Port Filter Options

In addition to the configurable fields, there are two options you can also use when you set up your filters. They are described in the sections below.

Pseudo Filter Option

Any of the combination port filters can be set to pseudo or real mode. In pseudo mode, the filter generates statistics, counting how many packets met the filtering criteria and sends an alarm to the NMS. It does not actually block any traffic.

The pseudo filter option provides a unique traffic monitoring capability. Uses of this capability include:

- Determining the effect a particular filter would have, without actually invoking it.

- Monitoring traffic patterns as an aid in determining optimum network design, usage policies, etc.
- Monitoring potential security threats.
- Evaluating security policies.

Values: either **Yes** (don't filter the packet; just count the packet for statistical purposes) or **No** (filter the packet if it meets the filtering criteria). **Yes** is the default.

Linking Combination Port Filters

Multiple combination port filters may be logically linked using the boolean and/or operators, as long as they apply to the same port and filter type. By default, each combination port filter represents a separate logical or test. That is, a packet will be filtered if the conditions specified in any of the filters are present.

When two or more combination port filters must match, you can use the **And** operator so the filters are linked with a logical **And** instead of a logical **Or**. A packet then will be filtered only if the conditions specified in both or all of the logically linked filters are present.

Values: either **Or** (filter based on the options specified just within this filter) or **And** (filter based on the options specified in this filter and the next filter for the same Port and Type). **Or** is the default.

5.6 ADDING A FILTER

You can add filters for any port. Every filter has many possible fields as described in the sections above. If all of the conditions in a filter are present within a packet, then the packet is filtered.

You can also add pseudo filters to any port. Pseudo filters allow you to gather statistics before you actually implement the filters. You can then use the statistics for “what-if” traffic and usage analysis.

Note: *If you are adding a filter to be used in conjunction with another filter and they must be ordered sequentially, use the **filters display** command to find the index number of the existing filter.*

Complete the following steps to add a filter or pseudo filter to a port. To accept a default value, just press Return:

1. Type: **filters add**

2. Enter the port number.

2 is the default. If the filter is for port 2, you don't need to enter anything; if the filter will be for another port, enter that number.

3. Select the filter type.

Entry is the default. If the filter will be an entry filter, you don't need to enter anything; if the filter will be an exit filter, type: **exit**.

4. Select whether the filter should be a true filter or a pseudo filter.

True is the default; meaning the filter will be a pseudo filter. You don't need to enter anything if the filter is to be pseudo. If you want the filter to be a true filter, type: **false**.

5. Select whether the filter will use a range of source MAC addresses.

NA is the default; meaning the filter will not use a source range. You don't need to enter anything if you are not using a source range. If you are not using a source range, go to Step 9.

If you are using a source range type either:

True – Filter the packet if the source MAC address is within the range.

False – Filter the packet if the source MAC address is outside the range.

6. Enter the first MAC address in the source range.
7. Enter the last MAC address in the source range.
8. Enter the source range MAC address mask.

ff:ff:ff:ff:ff:ff is the default address mask. If **ff:ff:ff:ff:ff:ff** is the mask you want to use, you don't need to enter anything. If you want to use a different mask, enter that value.

9. Select whether the filter will use a destination range of MAC addresses.

NA, is the default; meaning the filter will not use a destination range. You don't need to enter anything if you are not using a destination range. If you are not using a destination range, go to Step 9.

If you are using a destination range type either:

True – Filter the packet if the source MAC address is within the range.

False – Filter the packet if the source MAC address is outside the range.

10. Enter the first MAC address in the destination range.
11. Enter the last MAC address in the destination range.
12. Enter the destination range MAC address mask.
13. Select whether the filter will use a protocol match.

NA is the default. You don't need to enter anything if you are not using a protocol match. If you are not using a protocol match, go to Step 15.

If you are using a protocol match type either:

True – Filter the packet if the protocol type matches.

False – Filter the packet if the protocol type does not match.

14. Enter the protocol type to match.

15. Select whether the filter will use a field match.

NA is the default. You don't need to enter anything if you are not using a field match. If you are not using a field match, go to Step 20.

If you are using a field match type either:

True – Filter the packet if the masked value matches.

False – Filter the packet if the masked valued does not match.

16. Enter the field origin.

17. Enter the field offset.

18. Enter the field value.

19. Enter the field mask.

20. Select the operator.

Or is the default. You don't need to enter anything if the filter will use the **or** operator. If you want the filter to use the **and** operator, type **and**.

21. Enter the threshold time.

Zero (0) is the default. You don't need to enter anything if the filter won't use a threshold time. If you are not using a threshold, go to Step 23.

If you want the filter to use a threshold time, enter the value you wish to use.

22. Enter the threshold.

23. Enter the filter number

One (1) is the default. You don't need to enter anything if the filter number will be 1.

If you want the filter to have another index number, enter the value you wish to use.

LCM displays the filter you have just entered and prompts you whether you want to save it. Enter **y** (Yes) to save the filter or **n** (No) to cancel it. If you save the filter, it is redisplayed.

5.7 MODIFYING A FILTER

You can make changes to filters that you have already set up. You modify a filter in much the same way as you add a filter. LCM prompts you through each field. To modify a filter, begin with the steps below, then follow the prompts as if you were adding a filter:

1. Type: **filters modify**.

LCM prompts you for the filter index.

2. Enter the filter number.

LCM displays the filter type field. LCM prompts you through all of the filter fields the same as when you add a filter. What you had previously entered is displayed in brackets [] as the default value. Make any changes you wish following the instructions for adding a filter.

5.8 DELETING A FILTER

To delete a filter complete the following steps:

1. Type: **filters delete**.

LCM prompts you for the filter index.

2. Enter the filter number.

LCM responds filter deleted.

Note: All filter indexes are sequential, beginning with the number one. When a filter is deleted, all filters are renumbered so that the filter indexes remain sequential.

5.9 DISPLAYING A FILTER

To display a filter complete the following steps:

1. Type: **filters display**.

LCM prompts you for the port number.

2. Enter the port number.

LCM displays the filter which would look something like the filter display shown below, depending on how you had set up your filter:

```
Index: 1
Type: Entry
Pseudo: Yes
Source-range: True
Begin mac-add: 00:00:04:00:25:03
End mac-add: 00:00:04:40:15:03
Mask mac-addr: ff:ff:ff:ff:ff:ff
Dest-range: True
Begin mac-addr: 00:20:11:00:02:0a
End mac-addr: 00:40:02:15:22:01
Threshold: 5
Thres time: 3600
Pkt count: 179
Last source: 00:00:04:00:25:03
Operator: Or
```

5.10 FILTERING APPLICATION EXAMPLES

This section describes typical filtering applications which illustrate how a network manager can use the unique filtering abilities of the ATX to accomplish a variety of specific objectives. Specific capabilities illustrated by these application examples include:

- Selectively filtering network traffic for security purposes.

- Using a firewall filter to prevent problems and enhance performance.

For each application example, the situation is described first, and the objective to be accomplished is explained. Then, how the objective would be accomplished using the ATX is explained in general terms. In these examples, single letters are used to represent MAC-layer addresses. Real MAC addresses consist of a string of numbers, (22:14:15:4:5:6).

***Note:** The way that you configure filters will depend on the NMS you use. Instructions for using LCM to set up your filters are described in this chapter; refer to your NMS documentation if that is the tool you are using to set up your filter.*

5.10.1 Filtering for Security Purposes

Example 1 — Blocking access to a network segment

The objective in this example is to restrict access for security reasons. Workstations on one network segment (subnet) are to be restricted entirely from access to devices on an adjoining subnet.

In this example there are three subnets connected by a centrally located ATX (Figure 5-1). The subnets are referred to as the Engineering, Accounting and FDDI backbones.

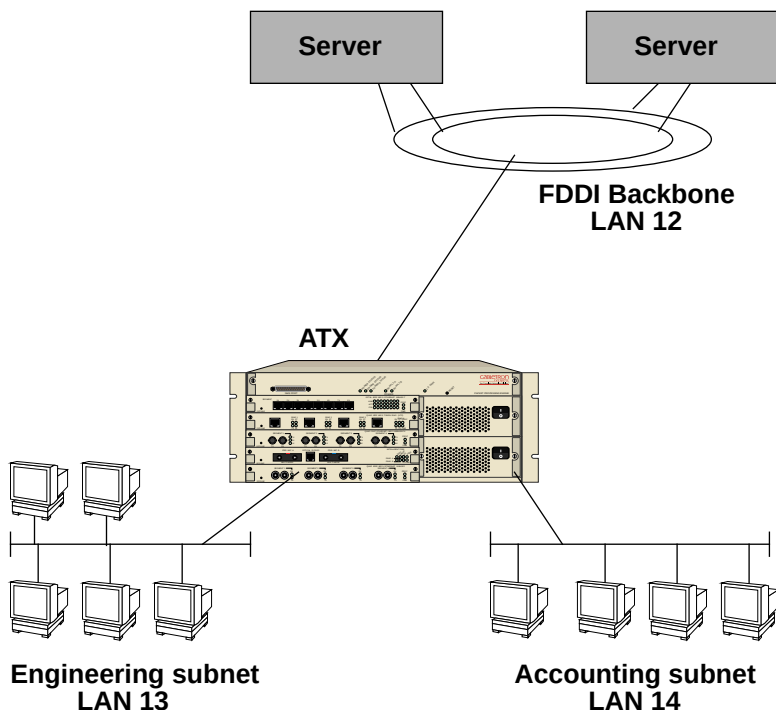


Figure 5-1. Using Filters To Restrict Access To An Adjoining Subnet

The company wishes to allow Engineering and Accounting workstations to access resources on the FDDI backbone, but wants to prevent Engineering users from accessing resources on the Accounting subnet. Therefore, the objective is to set up a filter that will block all traffic between LANs 3 and 4 while allowing users on both LANs (3 and 4) to access LAN 2.

For simplicity sake, assume that LAN 3 and LAN 4 are connected to the ATX's ports 3 and 4, respectively. LAN 2 is connected to the ATX's port 2.

Two combination port filters are used to discard any packets from Engineering destined for Accounting (LAN 3 to LAN 4) and any

packets from Accounting destined for Engineering (LAN 4 to LAN 3). Each filter includes:

- The source LAN or port number
- The destination port
- Match flags

The filters are constructed as follows:

- Filter 1: Identifier is port 4 as a destination
Fields are source LAN = 3, Match
- Filter 2: Identifier is port 3 as a destination
Fields are source LAN = 4, Match

Any packet whose source is LAN 4 and destination is Port 3 will be filtered. Likewise, any packet whose source is LAN 3 and destination is Port 4 will be filtered. However, the filters will not affect user access to the FDDI subnet (LAN 12). Therefore, the objective has been accomplished: LANs 3 and 4 (Engineering and Accounting) cannot interact, but users on either LAN can access LAN 2 (the FDDI backbone).

This is an example of logical segmenting. In this case, LANs 3 and 4 are distinct physical segments; however, before the filters were implemented, they were able to freely communicate. The filters were used to logically segment the network in such a way that LANs 3 and 4 cannot interact.

Example 2 — Blocking access to specific stations

A company uses a ATX to connect two LAN networks (Figure 5-2). Three computers on LAN 2 (the Accounting subnetwork) contain sensitive data (stations F, G, and H). The company wishes to prevent workstations on LAN 1 (Manufacturing) from accessing data on these three computers. Therefore, the objective is to prevent LAN 1 network users from accessing stations F, G, and H on LAN 2.

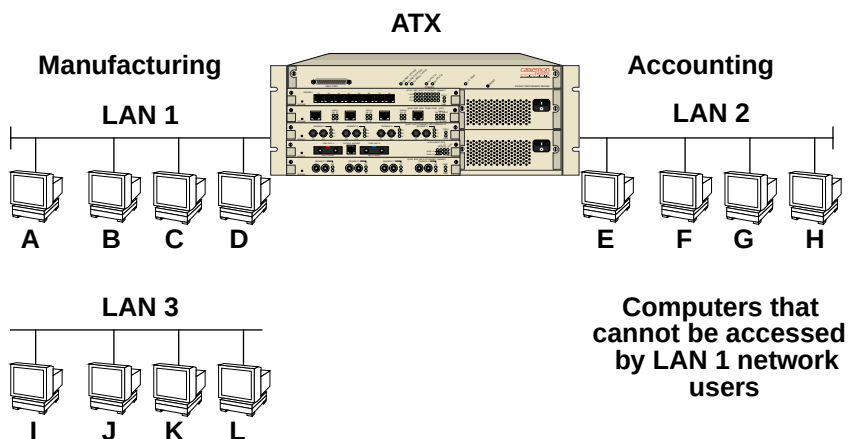


Figure 5-2. Using Filters To Restrict Access To Specific Workstations

In this example, a combination port filter is configured which instructs the ATX to discard data packets whose destination address is F, G, or H. Therefore, the ATX will not pass any packets from LAN 1 to LAN 2 if the packet's destination address is F, G, or H (the addresses of the computers containing sensitive data).

To accomplish the filtering objective, a combination port filter is required. This is because the filtering instruction must specify three separate components which cannot be included in an address table filter:

- Traffic from LAN 1
- Traffic destined for addresses F, G, and H on LAN 2
- Match flags for both components

This information is used to configure the filter as follows:

- Filter identifier – port number of the port attached to LAN 2 as a destination.

- Filter fields – destination address F-H (range, match) source LAN = 1 (match).

Note that a Match flag is specified for both fields; this instructs the ATX to filter any packets which match both fields (traffic from LAN 1 and to addresses F-H on LAN 2).

Several methods are available to accomplish this. For example, the combination filter could have been specified as follows:

- Filter identifier – port number of the port attached to LAN 1 as a source
- Filter fields – destination address F-H (range, match).

If LAN 3 did not exist, then the recommended approach is to use address table filters instead. Three filters (one for F, G, and H), should be created which specify filter all destination and filter all source.

This example is useful for illustrating three basic points concerning ATX filters:

- The example illustrates a paradoxical concept: even though an ATX is used to join network segments, it can also be used to block selected traffic, or all traffic if desired, between joined segments. The blocking mechanism is the filters you set up.
- Filters may be based upon various criteria: source address, destination address, packet type, etc. In the example just described, the filter criteria were source port and destination address.
- A filter can only block (discard) packets which must cross the ATX. The ATX in the example can only filter traffic that travels from LAN 1 to LAN 2 (or from LAN 2 to LAN 1). An ATX filter can prevent LAN 1 stations from accessing the sensitive-data computers on LAN 2 but cannot prevent station E from accessing these computers. The reason is that station E is on the same LAN as the sensitive-data computers and therefore does not need to use the ATX to access them.

Example 3 — Restricting access to authorized users

The example, shown in Figure 5-3, is very similar to the previous example. The difference is that access to stations F, G, and H will not be denied to all LAN 1 users. Instead, only authorized users on LAN 1 will be able to access the sensitive-data computers, stations F, G, and H on LAN 2.

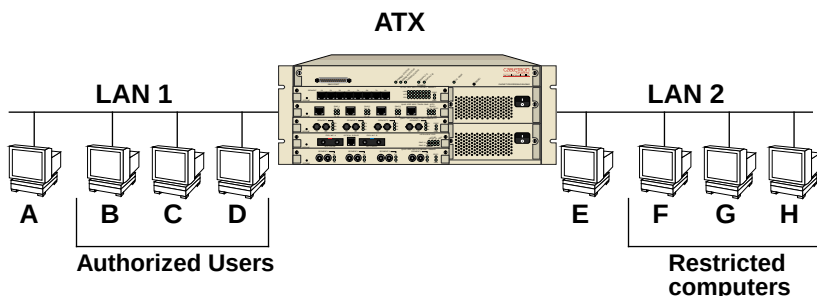


Figure 5-3. Using Filters To Restrict Access To Authorized Users

A combination port filter is configured that will allow data packets to be sent to the restricted computers (F, G, and H) only if the packet's source address is the address of an authorized user (station B, C, or D). The combination port filter's components are:

- Source addresses (of authorized users)
- Destination addresses (which identify packets directed to any of the restricted computers)
- No match flags for both of the above components

The filter is configured as follows:

- Source address field: B, C, or D (LAN 1), no match
- Destination address field: F, G, and H (LAN 2), no match

The No match flag is used in both fields to instruct the ATX to filter

all traffic that does not match both fields.

All packets destined for the restricted computers (F, G, or H) will be filtered unless the source address is the address of an authorized user (B, C, or D). Only authorized users will be able to access stations F, G, or H on LAN 2.

Note that the ATX is not storing information designed to identify restricted devices or authorized or unauthorized users. Instead it is using address information (which it does store) to act on filters which have been carefully configured to meet a desired objective: restrict access to certain devices to authorized users only.

Example 4 — Filtering by vendor ID

If you needed to know where all of the equipment from a particular vendor was located, you could set up a filter that would allow you to filter on the vendor ID. This could be useful if you need to find all the adapter cards from a vendor who had just released a new version of their driver software.

The first three bytes of a MAC address are always the vendor ID. If you want to filter by that ID, use the vendor ID followed by three octets of zeros and use `ff:ff:ff:00:00:00` as the mask. For example, Sun Microsystems's vendor ID is `08:00:20`, so you would use:

```
Source Range: [NA] (True/False/NA)> true  
Source Range Start: [00:00:00:00:00:00] >08:00:20:00:00:00  
Source Range End: [00:00:00:00:00:00] >  
Source Range Mask: [ff:ff:ff:ff:ff:ff] >ff:ff:ff:00:00:00
```

You could then set a threshold, so that an alarm would be sent every time a packet with that vendor ID was processed. The alarm would include the MAC address of the originating device. Over a fixed amount of time, the probability is high that you would receive an alarm from every device from the specified vendor (which would include some duplicates).

Note: In order for this trap to work, you must have `ConfigAlarmDynamic` set and your NMS must be able to process traps from the ATX.

Example 5 — Configuring a firewall filter to control multicasts

To optimize network performance, you can configure filters to reduce multicasts (packets broadcast to multiple destinations). Furthermore, you can prevent multicasts of packets of a particular protocol type. In this example, four LANs are interconnected by an ATX (Figure 5-4). The objective is to prevent LAN 1 from sending AppleTalk I multicasts to LANs 2 and 3, yet allow AppleTalk I multicasts from LAN 1 to LAN 4. The filter described below is a firewall filter; it acts as a barrier to protect the network from a condition that may degrade network performance.

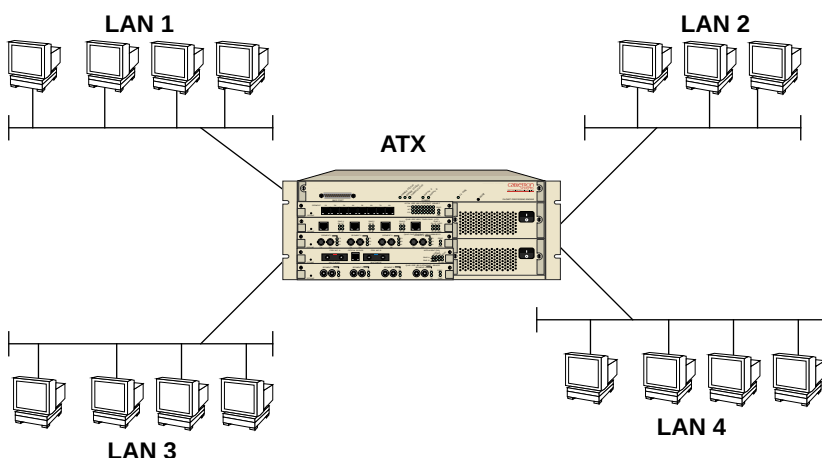


Figure 5-4. Using Firewall Filters To Reduce Multicasts

This filter is configured as follows:

- Filter identifier – port number of the port attached to LAN 4 as a destination
- Filter fields – protocol type = AppleTalk I, match source LAN = LAN 1, match destination address = 01-00-00-00-00-00 with mask 01-00-00-00-00-00, match

This filter will block AppleTalk multicasts (or all AppleTalk traffic if the destination address field is omitted) from LAN 1 to LANs 2 and 3, yet AppleTalk I traffic to LAN 4 will be permitted (because LAN 4 is not specified for filtering).

CHAPTER 6

TRAPS

The ATX sends trap PDUs to an SNMP Manager, using a pre-configured SNMP Manager IP address. (See `configNMSAddress` in the *ATX MIB Reference Guide*). If no address has been pre-configured, then the ATX sends the traps to the source IP address of the last SNMP datagram received from an SNMP Manager. If no address has been pre-configured, and if no datagrams have been received since the ATX was booted, then the ATX uses the broadcast IP address.

The trap PDUs are sent from source UDP port number 161, to destination UDP port number 162, which are the SNMP standard numbers reserved for Trap PDUs. The ATX may be configured to send an additional copy of each Trap PDU to a UDP port number you specify. (See `sysTrapPort` in the *ATX MIB Reference Guide*.)

6.1 GENERIC SNMP TRAPS

- `coldStart` (0) – An ATX has re-started, and the SNMP Manager's copy of the ATX's configuration may be incorrect.
- `warmStart` (1) – Not used by the ATX.
- `linkDown` (2) – A port has failed, and the ATX's local management agent has disabled usage of the port. The variable bindings portion of the trap contains the `ifIndex` of the port.
- `linkUp` (3) – A port has come back to life, and the ATX's local management agent has re-enabled usage of the port. The variable bindings portion of the trap contains the `ifIndex` of the port.
- `authenticationFailure` (4) – This trap is generated whenever the community name in a PDU does not match the corresponding password. All `SetRequest` PDUs must match the `configAnyPass` (refer to the description of `configGetPass` for `SetRequest` exceptions), `GetRequest` PDUs must match either the `configGetPass` or the `configAnyPass`. The `GetRequest` exception is

for one of the debugging attributes; those PDUs must always provide the `configAnyPass`.

- `egpNeighborLoss` (5) – Not used by the ATX.
- `enterpriseSpecific` (6) – The ATX is reporting some interesting information, which is contained in the variable-bindings portion of the PDU. If the ATX has been configured to require acknowledgments to its Trap PDUs (`sysTrapAck`), then the SNMP Manager must acknowledge the trap, generally by issuing a `GetRequest` for the significant variables involved in generating the trap.

All enterprise specific traps specify the value of `sysObjectID.0`, the object identifier `sysID`, as the enterprise, and zero as the specific-trap. The variable-bindings may contain many pairs of variable names and their values.

***Note:** These traps are only sent if `ConfigAlarmDynamic` is set.*

The ATX can generate enterprise specific traps in either an Acknowledge or a Non-Acknowledge mode. In the Non-Acknowledge mode, a single trap PDU will be sent to the SNMP Manager. This PDU will contain sufficient information for the SNMP Manager to determine the cause of the trap in Acknowledge mode. The ATX resends the PDU occasionally (see `sysTrapTime` and `sysTrapRetry` in the *ATX MIB Reference Guide*) until it receives an Acknowledgment from the SNMP Manager. The Acknowledgment is defined to be a `GetRequest` for the variables associated with the Trap's cause. Note that Acknowledge mode is only applicable for enterprise specific traps; that is, only enterprise specific traps may be resent.

Many traps include other variable bindings, in addition to the variable binding which caused the trap to be generated. The additional variable bindings are not guaranteed to be present, and/or may be passed in separate trap PDUs.

6.2 ATX UNIQUE TRAP IDS

The ATX possesses unique trap IDs which allow a SNMP Manager (Spectrum Element Manager, Spectrum) to have more control over SNMP Traps. Each trap is given a unique Trap ID, which gives detailed information about the trap and why it was sent. This also gives you the ability to select the traps you want generated and the traps you want to suppress.

- **Tempok (1)** - Sent whenever the module's temperature transitions from too hot to okay, and vice versa.
- **Writestatus (2)** - Sent when a bank of Flash EPROM has been erased. If `swdisWriteStatus` indicates success, then the unit is ready to be downloaded with the new software.
- **PortFunctions (3)** - Sent whenever the current functional state (active protocols) of the port has changed.
- **RxQueues (4)** - Sent whenever the number of times that the port's receiver has stopped receiving packets due to buffer space shortages has exceeded the port's limit.
- **TxStormFlag (5)** - Sent whenever multicast storm protection has been invoked for the port.
- **TxCongests (6)** - Sent whenever packets destined for the unit itself were discarded due to lack of buffer space.
- **filterThresh (7)** - Sent whenever usage of a port's combination filter has exceeded the filter's limits.
- **debugStringId (8)** - Send whenever the unit has a debug text string to be displayed. The text strings are sent in a stream-like fashion.
- **IpbkOperation (9)** - Send whenever the unit has finished a loop back test, or a loop back error has been detected.

- `trunkState` (10) - A trunking state change transition has occurred. The possible transitions are:
- CLOSED - ONEWAY
- ONEWAY - PERTURBED
- PERTURBED - JOINED
- JOINED - HELDDOWN
- CLOSED - HELDDOWN
- ONEWAY - HELDDOWN
- PERTURBED - HELDDOWN

`trunkBridgeAddr` (11) - The associated trunking MAC address of the bridge ID of the remote bridge has changed.

`trunkIPAddr` (12) - The associated trunking IP address of the remote bridge has changed.

`trunkError` (13) - An error has occurred in trunking.

`trunkLinkOrdinal` (14) - The port's index in the trunking group has changed.

`trunkLinkCount` (15) - The number of ports in the trunking group has changed.

`diagUnitBooted` (16) - The unit has booted.

`storageFailure` (17) - Sent if the unit's Configuration EEPROM has failed. The unit will not be able to reboot, and must be returned to the factory.

`portCongested` (18) - Sent whenever outbound congestion control has been invoked for the port.

`topChangeBegun` (19) - The spanning tree topology has begun to change.

`topChangeEnd` (20) - The spanning tree topology has stopped changing.

`ifErrors` (21) - Sent whenever the number of hardware errors in received and transmitted packets has exceeded the port's limit.

`stRootID` (22) - The spanning tree root bridge ID for the unit has changed.

`stRootCost` (23) - The unit's spanning tree cost to the root bridge has changed.

`stRootPort` (24) - The unit's spanning tree root port has changed.

`stMaxAge` (25) - The unit's spanning tree maximum age has changed.

`stHelloTime` (26) - The unit's spanning tree hello time has changed.

`stForwardDelay` (27) - The unit's spanning tree forward delay time has changed.

`stDesigRoot` (28) - The Root Bridge ID in received Spanning Tree Configuration BPDUs from the port has changed.

`stPortDesigBridge` (29) - The bridge ID of the spanning tree designated bridge of the LAN/WAN to which the port is attached has changed.

`stPortDesigCost` (30) - The cost to the spanning tree root bridge from the designated port of the LAN/WAN to which the port is attached has changed.

`stPortDesigPort` (31) - The port ID of the spanning tree designated port of the LAN/WAN to which the port is attached has changed.

`stPortState` (32) - The spanning tree state of the port has changed.

fddimibSMTCFState (200) - Sent whenever the FDDI port's CFM state has changed. The **fddimibPORTMACIndicated** (one or two instances, depending upon whether the FDDI connection is a SAS or a DAS) is also included.

fddimibMACUpstreamNbr (201) - Sent whenever the FDDI port's upstream neighbor has changed.

configPowerAc1 (202) - Sent whenever the AC input to the unit's first power supply transitions from on to off, and vice versa.

configPowerAc2 (203) - Sent whenever the AC input to the unit's second power supply transitions from on to off, and vice versa.

configPowerDc1 (204) - Sent whenever the DC output of the unit's first power supply transitions from on to off, and vice versa.

configPowerDc2 (205) - Sent whenever the DC output of the unit's second power supply transitions from on to off, and vice versa.

configPowerPresent1 (206) - Sent whenever the presence of the unit's first power supply transitions from present to not present, and vice versa.

configPowerPresent2 (207) - Sent whenever the presence of the unit's second power supply transitions from present to not present, and vice versa.

sfddiShortAddressing (208) - Sent whenever an FDDI packet is received (other than Claim/Beacon packets) with 16 bit MAC addresses.

sfddiSmtConditionsStatus (209) - Sent whenever any of the conditions indicated by the value of the MIB variable **sfddiSmtConditions** has occurred.

sfddiSrfConditionsStatus (210) - Sent whenever any of the conditions indicated by the value of the MIB variable **sfddiSrfConditions** has occurred.

sfddiSBFlag (211) - Sent whenever the FDDI port's optical bypass becomes stuck, or un-stuck.

`sfddiOBSFuseBad` (212) - Sent whenever the fuse to the FDDI port's optical bypass becomes bad, or switches from bad to good.

`sfddiStationState` (213) - Sent whenever the FDDI port's Station State has changed.

`swanActualSpeed` (214) - The actual line speed of the WAN port has changed.

`fddismtUpstreamRsp` (215) - The upstream neighbor of the requested FDDI device has been learned.

`hwFatalErr` (216) - Sent whenever a module dies unexpectedly. Since death of the ME or TURBO causes the unit to reboot, a `hwFatalErr` alarm for such a module will never occur.

`hwDiagModuleFailed` (217) - A module has failed diagnostics.

`hwDiagModMismatch` (218) - The type of a module does not match its defined type.

`hwDiagPortMismatch` (219) - A type of a port does not match its defined type.

`hwDiagPortStatus` (220) - A port of this module has a bad status.

`sfddiDup` (221) - Duplicate MAC address found on FDDI ring.

`slog` (222) - An event logging message has been generated.

`atRtOvflw` (223) - Appletalk routing table overflow.

`atArpOvflw` (224) - Appletalk ARP table overflow.

`sipxRtOvflw` (225) - IPX routing table overflow.

`sipxSvcOvflw` (226) - IPX service table overflow.

`atPortsSameSeg` (227) - Appletalk- two ports on same segment.

`iprouteTbleOvflw` (228) - Routing table overflow.

`arpTbleOvflw` (229) - ARP table overflow.

eePromReconfig (230) - The unit's EEPROM has been reconfigured.

maxNextHop (231) - Maximum number of next hops reached.

ripBadNet (232) - RIP received with wrong local network number.

routeAgeOut (233) - Route aged out.

sipxSAPAgeOut (234) - IPX service aged out.

ipUnknownDest (235) - IP packet to unknown destination received by host.

pppLinkOpen (236) - PPP link to open

pppLinkClose (237) - PPP link to close.

pppNeighborIpAddrChange (238) - PPP neighbor IP address change.

dupIP (239) - Duplicate IP address detected.

CHAPTER 7

DIAGNOSTICS AND TROUBLESHOOTING

The main topics covered in this chapter are:

- Power-up diagnostics
- Diagnostics while the ATX is operational
- Status and activity indicators (LEDs)
- Troubleshooting

7.1 DIAGNOSTICS OVERVIEW

The ATX incorporates several built-in diagnostic and testing capabilities which are convenient to use and cause minimal or no disruption to the operational network. These capabilities are effective for isolating problems within the ATX. Built-in diagnostic capabilities include:

- System-wide power-up diagnostics, which are run every time the system is powered up or reset.
- Local and remote loopback tests, which can be performed on each interface module.
- Temperature sensors on each module (described in the individual module manuals).
- Externally visible status indicators on each module (LEDs).
- Status of power supplies.

All tests can be performed locally or remotely using an in-band or out-of-band NMS.

7.2 POWER-UP DIAGNOSTICS

The ATX performs an extensive set of diagnostic self-tests whenever any of the following events occurs:

- Power-up
- Reset using the front panel reset button
- Reset via the NMS (a soft reset)
- Automatic reset occurs in response to a non-recoverable failure

The power-up diagnostics test processors, memory, and other critical components on all ATX modules. Power-up diagnostics also verify proper interaction between all system modules.

The processors on the PPE are tested first, and then each of the interface modules is tested in order (from top to bottom of the unit). LEDs on the front panel indicate progress of the power-up diagnostics sequence and the status of each module as determined by the power-up diagnostics.

If an interface module fails a critical test, it is automatically disabled. When the power-up diagnostics are completed (within 60 seconds after power is applied), the main element module gathers the test results and generates and sends appropriate traps (alarms or event messages) to the NMS.

All diagnostics software is stored in nonvolatile memory (EPROM).

7.2.1 Power-up LED Sequence

When you power up your ATX, the following occurs:

1. All LEDs turn on briefly (this does not apply to all 10 Mbps Ethernet models, refer to the individual modules Users Guide).
2. Individual module LEDs become active, starting with the LEDs on the PPE and continuing downward until all the modules have completed power-up diagnostics.
 - a. The POWER STATUS, ENGINE STATUS, and POWER SUPPLY LEDs on the PPE and the POWER LEDs on the

modules are on for approximately 3 seconds.

- b. The ENGINE STATUS LED on the PPE begins to flash.
 - c. The ENGINE STATUS LED continues to slowly flash while the remaining modules are running power diagnostics.
 - d. The TURBO STATUS LED stays on for approximately 3 seconds; then it flashes.
3. After the last interface module has completed its power-up diagnostics the Packet Processing Engine's ENGINE STATUS LED will stay on solidly.
 4. The TURBO STATUS LED will come on, followed by the STATUS or PROC LEDs of the interface modules (from the top down).
 5. The LEDs will indicate that the ATX has begun proper operation, as shown in Figure 7-1.

7.2.2 Specific Power-up Tests

The power-up diagnostic tests performed on the PPE are:

- ROM checksum test
- VDRAM memory tests
- Memory map tests and interrupt tests
- Shared RAM component – AMD 29000 processor's access to shared RAM, DMA controller's access to shared RAM, various DMA functions, DMA / AMD 29000 shared RAM arbitration, interface modules' access to shared RAM and arbitration with simultaneous access by all processors.

For power-up tests performed on individual modules, refer to the module's manual.

7.2.3 Software Checksum Comparison

When the ATX reboots, its operational software is verified by a checksum comparison before it is loaded. If the software fails the checksum test due to an aborted new software distribution procedure, the ATX will automatically use its backup version of software. (A backup version of software is always stored in nonvolatile memory.)

The operational parameters of the ATX software are also protected by a checksum comparison. When the ATX reboots, if the operational parameters of the ATX fail a checksum test due to a power failure in the midst of a previous update, the ATX will automatically use its backup version of the parameters. (A backup version of the operational parameters is always stored in nonvolatile memory before any update is attempted.)

7.2.4 Power-up Diagnostics Results

After completion of the power-up diagnostic sequence, all STATUS LEDs on the ATX front panel should be on, indicating that the modules have passed the power-up tests.

7.2.5 Responses to Failures at Power-up

How the ATX responds to failures detected during power-up depends on the seriousness of the failure. For example, a failure detected on an interface module will not prevent the ATX from booting; it will only cause the particular interface module to go out of service. If an interface module fails power-up diagnostics, it is automatically disabled. The rest of the ATX system will continue to operate. Similarly, the ATX will operate if a non-critical component such as the out-of-band management port on the PPE fails diagnostics.

However, in the event of a critical failure such as a failure of the main element processor, local RAM or SRAM, the ATX will halt execution and will not boot to operational mode.

Failure Indicators

If an FDDI or Ethernet module has failed, its front panel STATUS LED will be off.

NMS Failure Traps

As each module completes its power-up diagnostics, it saves information on any detected failures. These results are passed to the NMS when the power-up diagnostics are completed (assuming the ATX is operational). The results sent to the NMS indicate which component has failed.

7.3 DIAGNOSTICS WHILE ATX IS OPERATIONAL

The following diagnostics may be performed while the ATX is operational:

- Local and remote loopback tests on any port.
- Power-up diagnostics on an Ethernet or Token Ring module.

7.3.1 Loopback Tests

Built-in local and remote loopback tests can be used to test individual ports while the ATX is operational. When in local loopback, a port is disconnected from its network. The ATX generates loopback packets for the port, and the port loops the packets back without sending them onto its network.

During a remote loopback test, the port is in normal operation, sending, and receiving packets to its network. The ATX generates loopback packets which are sent out of the port to a particular destination device on the port's network. The destination device echoes the packet back onto the network, and the originating port receives the packet.

For both types of tests, normal operation is indicated when generated packets are received back without errors. For remote

loopback tests, the ATX creates LLC Type 1 test packets for LANs, and PPP echo-request packets for WANs and UARTs.

Both types of loopback tests can be initiated by the NMS, and test results are reported to the NMS. Refer to the *ATX MIB Reference Guide* for the MIB variables.

7.3.2 Diagnostic Results

ATX diagnostic results are indicated in two ways:

- By observing the front panel LEDs.
- By reading NMS trap messages.

Both power-up and loopback diagnostics produce traps, which are sent to the NMS and may be logged for future reference. In some cases it may be more convenient to simply observe the LEDs, but in most cases traps provide more information. There are no LEDs for the loopback tests; the results of these tests must be observed by accurate packet transmission, or read by using an NMS to examine traps.

7.4 STATUS AND ACTIVITY LEDS

The front panel of the ATX includes numerous LEDs which show the status or activity of various system components.

During normal operation, the status of each LED (off, on, or flashing) should be as shown in Figure 7-1. The meaning of the ATX base unit LEDs is explained in Table 7-1. The meaning of the module LEDs are explained in the individual module manuals. (For LED activity during power-up, See Chapter 2, **Power-up Diagnostics Sequence**.)

Table 7-1. Meaning Of ATX LEDs

LED	Meaning
POWER STATUS	On – Power is on and the voltage is within the acceptable range.
ENGINE STATUS	On – Packet Processing Engine is ready for operation. Blinking – A module is overheating.
TURBO STATUS	On – Turbo (a key packet processing component on the Packet Processing Engine) is ready for operation.
POWER SUPPLY A	On – Power supply A is generating sufficient voltage for the ATX to operate. Off – Power supply A is not present, switched off, or malfunctioning.
POWER SUPPLY B	On – Power supply B is generating sufficient voltage for the ATX to operate. Off – Power supply B is not present, switched off, or malfunctioning.
1.6 Gbps	On – Indicator that this ATX has a 1.6 Gbps backplane.

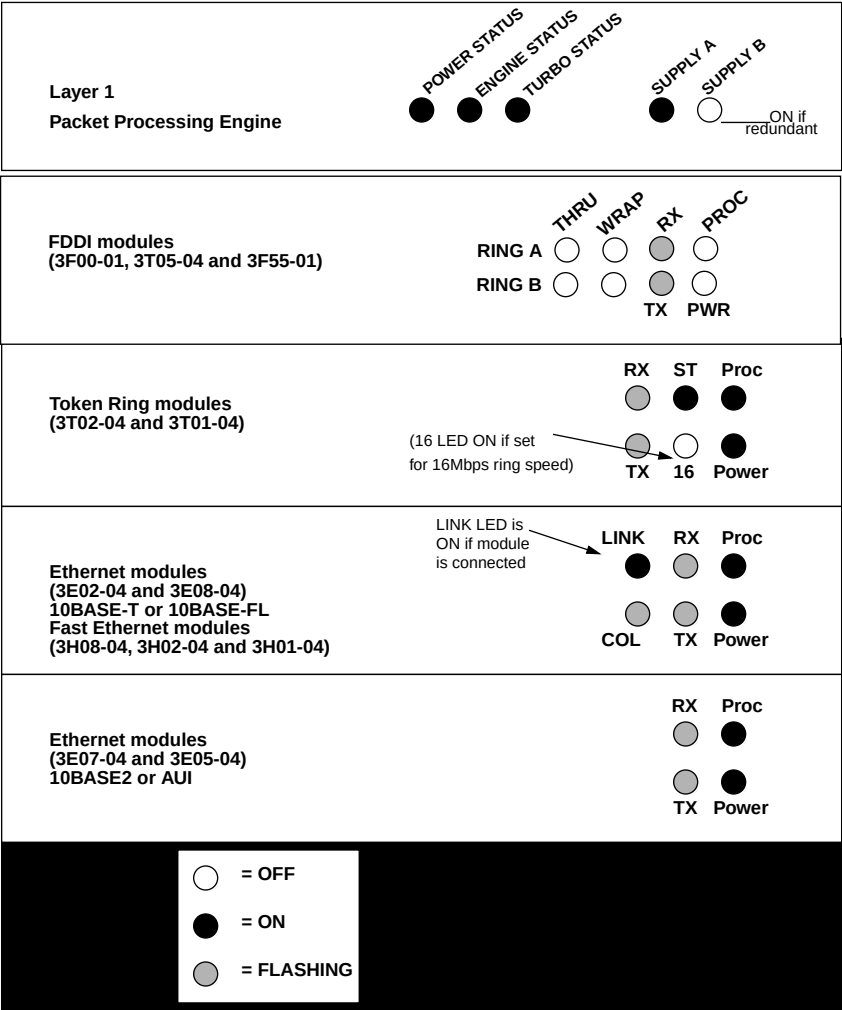


Figure 7-1. LED Activity During Normal Operation

7.5 TROUBLESHOOTING

This section lists several problem situations that could be encountered while using the ATX and suggests appropriate action.

Because every situation is potentially unique and may involve unique external factors, the corrective actions suggested here should be considered as guidelines only.

7.5.1 ATX Does Not Power Up

If your ATX does not power up, check each one of the following; if it still doesn't power up, contact Cabletron Systems Technical Support.

- Make sure the power source is operational.
- Make sure the power cord is securely connected.
- Make sure the power supply switch is on.
- Check the power supply fuse.

7.5.2 Module Status LED Not Lit

If the Module Status or PROC LED is off on one of your I/O modules, follow the steps below:

1. Check the status of ports using LCM.
2. Restart the module.

The module may be restarted by removing and re-inserting the module, or by taking the module offline and online using LCM.

For more information, refer to the individual module manuals.

7.5.3 Connectivity Problems

- Check for LED abnormalities, which may help indicate the source of the problem.
- Check status of ports using LCM.

- Check for loose port connections. Check all connectors to the modules (especially twisted pair connectors, which may be fragile).
- Check to make sure all the modules are firmly connected; check all the screws are fully tightened.
- Number of carrier losses is increasing. This indicates that the transceiver is not present or the transceiver's 10BASE-T connection is suspect. Check the transceivers to ensure they are firmly connected.
- Number of total collisions has dramatically increased. Check the BNC transceiver, it may be lacking a terminator or its BNC connector may be touching another transceiver's BNC connector. Packets that are not being bridged/routed correctly can also cause total collisions to increase. Verify the aging time and the IP addresses of the ATX.
- Make sure UTP Transceivers are not pre-10BASE-T.
- DEC Broadband AUI Transceiver is not supported by the ATX.
- Unix spray command loses packets if workstations are busy.
- RESET button is depressed in an effort to reset the module. The RESET button merely takes the module offline. To reset the module or to bring it online use LCM. (On newer modules the button is labelled OFFLINE, rather than RESET.)

7.5.4 ATX Has Rebooted

- Check the NMS configFatalErr trap to determine the reason for the reboot.

7.5.5 ATX Does Not Respond To NMS

- Check status of ports using LCM.
- See if Spanning Tree topology is stable on LCM.

- Check that a pathway to the ATX exists (intermediate bridges and routers are functioning).
- Verify ATX's IP addresses, one at a time using LCM.
- Verify values of `configNMSAddress`, `configAnyPass`, and/or `configGetPass`.

CHAPTER 8

ADDING/SWAPPING MODULES AND MAINTENANCE

The ATX configuration may include a total of five interface modules in various combinations. This means any configuration that does not use all five interface slots may be expanded by installing additional interface modules. Install the additional interface module in any vacant interface slot.

Caution: *Observe all Electrostatic Discharge (ESD) precautions before handling the ATX. Failure to do so could result in damage to the ATX and other associated components.*

8.1 ADDING A MODULE

If the module you are adding is of a different type than the module that previously occupied that slot, or if the slot was previously vacant, you have to reboot the ATX so it will recognize the new module.

If the module you are adding has a different number of ports than the module you are removing, before you can power cycle the ATX, you must:

- Delete all static addresses
- Delete all ARP addresses and IP routing table entries
- Delete all filters

Follow these steps to install an additional interface module:

1. If the module is still online, take it offline using the LCM offline command.

To see if the module is online, check its PROC LED; if the module is online, the LED will be lit.

2. Remove power from the ATX.
3. Disconnect the module from the network.

4. Loosen the screws at each end of the panel that covers the interface slot and remove the protective panel.
5. Gently slide the module into the plastic guides in the module slot until it is completely inserted.
6. Push the module firmly into place to fully engage the connectors at the back of the module with the backplane at the rear of the ATX chassis.
7. Tighten the screws on each side of the module's front panel.
8. Power cycle the ATX, and check the module's LED power-on sequence as described in manual accompanying the module.
9. Connect the module to the network as described in the manual that accompanied the module.
10. Use LCM or an NMS to add back any filters, static addresses, ARP addresses, and IP routing table entries that you deleted.

8.2 SWAPPING A MODULE

To replace modules of the same type perform the following procedures.

Caution: *Do not remove an interface module if its STATUS LED is on. If the STATUS LED is on, press the module's OFFLINE (or RESET) button, use LCM to take the module offline, or use an NMS to disable the module.*

To hot swap an interface module, complete these steps:

1. Take the ATX offline by pressing the OFFLINE button, use the LCM to take the ATX offline, or use an NMS to disable the module.
2. Loosen the screws at each end of the front panel of the interface module you are going to replace.

3. Remove the installed interface module by pulling gently but firmly on the “ears” at the ends of the module's front panel.
4. Gently slide the new module into the plastic guides in the module slot until it is completely inserted.
5. Push the module firmly into place, as far as it will go, to fully engage the connectors at the back of the module with the backplane at the rear of the ATX chassis.
6. Tighten the screws on each side of the interface module's front panel.
7. Connect the module to the network as described in the manual that accompanied the module.
8. Bring the module online using LCM or an NMS if you took it offline using LCM or an NMS.

8.3 MAINTENANCE

You may need to check the fuse in your ATX or to clean the filters on the fan. Those procedures are described in the sections that follow.

8.3.1 Power Fuse

Each power supply contains a 6.3 ampere 250 V slow-blow fuse, located immediately above the three-prong power input connector on the back of the ATX. If you think this fuse may have blown, inspect it for visible damage and /or replace it. In most cases, any damage to the fuse will be readily apparent (e.g., shattered fuse, blackened glass, broken fuse element). If you see no damage but suspect a fuse problem, try replacing the fuse.

Warning: *Do not remove the fuse from the ATX while power is applied to the unit.*

To access the fuse:

1. Disconnect the power cord from the ATX.
2. Pull the small plastic fuse drawer below the power input connector directly outward.
3. Remove and replace the fuse.

Caution: *For continued protection against fire hazard, replace only with 250V slow-blow 6.3 amp fuses.*

4. Push the fuse drawer back into the power input filter housing until it snaps into place.

8.3.2 Fan Filters

Each ATX comes equipped with three fans located in the back of the unit. The screens covering these fans need to be cleaned on an annual basis to prevent overheating. You don't need to remove the fans or screens to clean them. Use any spray non-residue dust remover to remove the dust.

Note: *If your environment is dirty, the filters may need to be cleaned more frequently.*

8.3.3 Hot Swapping the Power Supply

It is critical that the power supply inserted into the top slot of the ATX chassis be installed very carefully if you are installing it while the ATX is powered on. Failure to use caution while installing the power supply could cause it to come in contact with the bottom of the Packet Processing Engine (PPE) board, causing the PPE to short circuit.

Warning: *Removing the ATX Power Supply must be performed by qualified service personnel. Failure to follow all instructions and warnings may result in an electric shock hazard.*

The power supply must slide straight into the chassis underneath the tabs shown in black in Figure 8-1 and rest on its metal support bracket, also shown in Figure 8-1. If you attempt to slide the power

supply into the chassis at an angle, or if you position the power supply above the tabs shown in Figure 8-1, you risk short circuiting the PPE board.

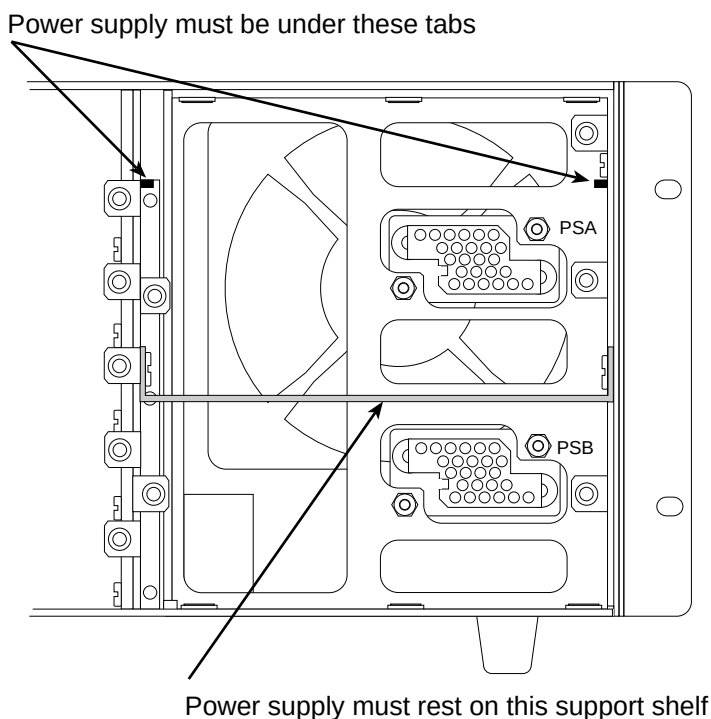


Figure 8-1. Chassis With Power Supply A Positioning Tabs And Supporting Shelf Indicated

To replace the power supply in slot A (the top slot):

1. Turn power switch on Power Supply A (PSA) off.
2. Remove the two thumb screws holding the power supply in place.
3. Pull the power supply straight out.

4. Slide the new power supply straight into the chassis under the tabs shown in Figure 8-1.

The power supply should be placed as shown by the dotted line rectangle in Figure 8-2.

5. Tighten the two screws that hold the power supply into the chassis.
6. Turn the PSA power switch on

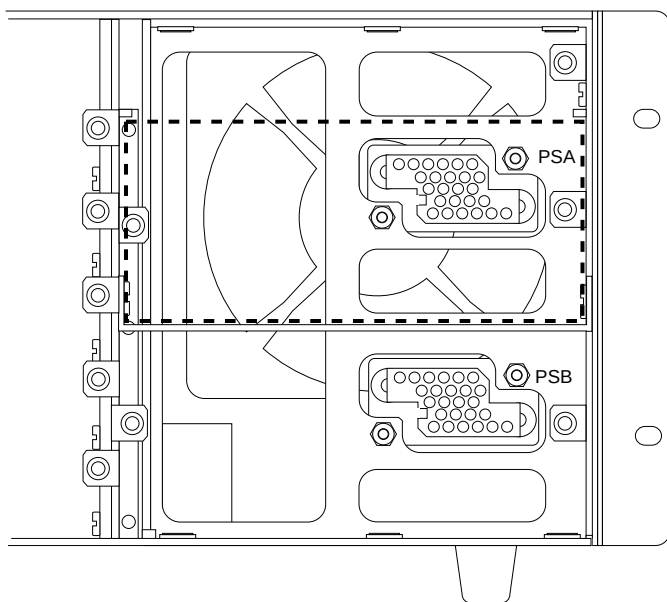


Figure 8-2. ATX With Power Supply A Position Indicated

APPENDIX A

SPECIFICATIONS FOR THE ATX

A.1 PACKET PROCESSING ENGINE

Dual AMD 29000 RISC processors

4 MB FLASH memory

8 MB main memory

2 MB shared memory

128 KB configuration memory

1.6 Gbps internal bandwidth

A.2 STANDARDS COMPLIANCE

A.2.1 Protocols

- ANSI FDDI X3T9.5 (SMT 7.3/MAC-2)
- IEEE 802.1d, 802.2, 802.3, 802.5, 802.5m
- RFCs for IP, UDP, ICMP, ARP, PPP, RARP, Proxy ARP, RIP, IP packet fragmentation (791), MIB II (1213), SNMP

A.2.2 Switching Modes

- Transparent Bridging, 802.1D Spanning Tree (Ethernet, Token Ring, FDDI, HSSI)
- Source Routing (Token Ring), IBM Spanning Tree
- Source Routing Transparent (Token Ring, FDDI, HSSI)
- Token Ring Translations (IP, IPX, AppleTalk, NetBios, SNA)
- Source Route Translational Bridging

A.2.3 Local Routing

- IP Routing (RIP)
- AppleTalk Routing
- IPX Routing (RIP, SAP, Diagnostic)
- IP Multicast Support (DVMRP)

A.2.4 Interfaces

- EIA
- RS-232C

A.3 PHYSICAL (BASE UNIT)

Height	7.0 in. (17.78 cm)
Width	16.8 in. (42.67 cm)
Depth	18.0 in. (45.72 cm)
Weight	31.25 lb. (14.20 kg)
Installation options	Tabletop or rack-mount

A.4 PHYSICAL (POWER SUPPLY)

Height	2.6 in. (6.60 cm)
Width	6.7 in. (17.02 cm)
Depth	14.6 in. (37.08 cm)
Weight	7 lb. (3.20 kg)

A.5 ELECTRICAL

Input voltage	Auto-ranging from 100 to 120 or 200 to 240 Vac
Frequency	47 to 65 Hz
AC power	380 W
Maximum AC Current	
Requirements	4 amps – 100 to 120 Vac 2 amps – 200 to 240 Vac

A.6 ENVIRONMENTAL

Operating temperature	5° C to 40° C (41° F to 104° F)
Relative humidity	0% to 95%, non-condensing
Storage temperature	-30° C to 90° C (-22° F to 194° F)

A.7 SLOTS

I/O module slots	Five
------------------	------

A.8 OUT-OF-BAND CONNECTOR

RS-232C using Point-to-Point (PPP) or Local Console Manager (LCM)

A.9 DIAGNOSTIC LEDS (BASE UNIT)

Power status

Engine status

Turbo status

Power supply A

Power supply B

Reset

A.10 SOFTWARE LOADING

FLASH memory via TFTP

A.11 ADDRESS TABLE SIZE

8,192 dynamic (learned) entries default, expandable to 16,384

A.12 CERTIFICATION

Safety	UL 1950, CSA C22.2 950, EN 60950, and IEC 950
Emission	FCC Part 15 Class A, EN 55022 Class A, and VCCI Class I.
Immunity	EN 50082-1

APPENDIX B

PACKET TRANSLATION PROCEDURE

Since the ATX is a multi-media unit, packets are converted from the different media into a standard canonical format. The **Offset** field for the **filters** command refers to the canonical format packet. The exact translation procedure is defined by RFC 1188 and RFC 1042, except for the encapsulation of Ethernet AppleTalk packets which uses Protocol ID of 00-00-F8 instead of all zeros. For further information, refer to the RFCs, or to the figures which follow.

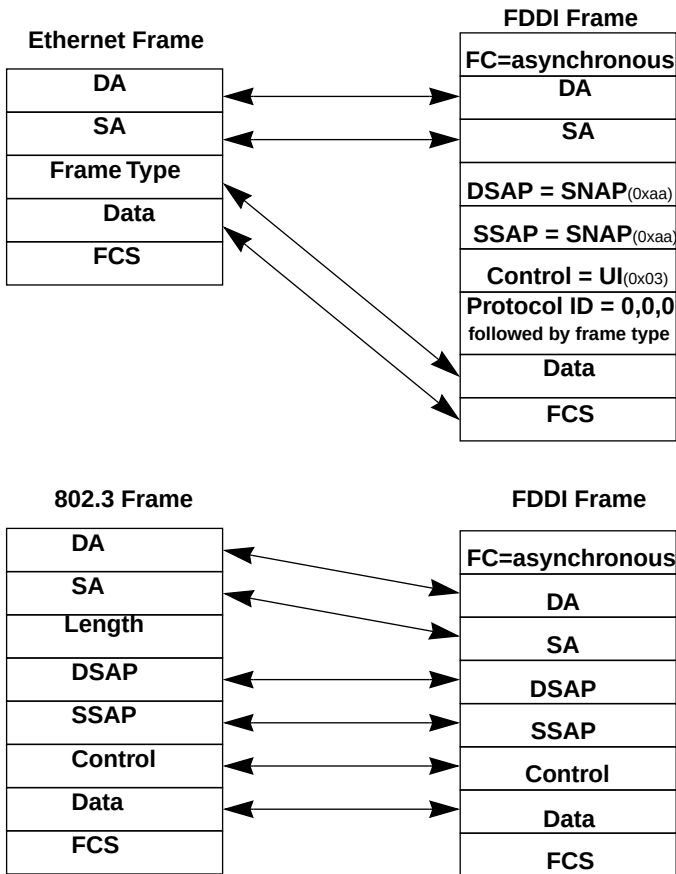


Figure B-1. Packet Translation

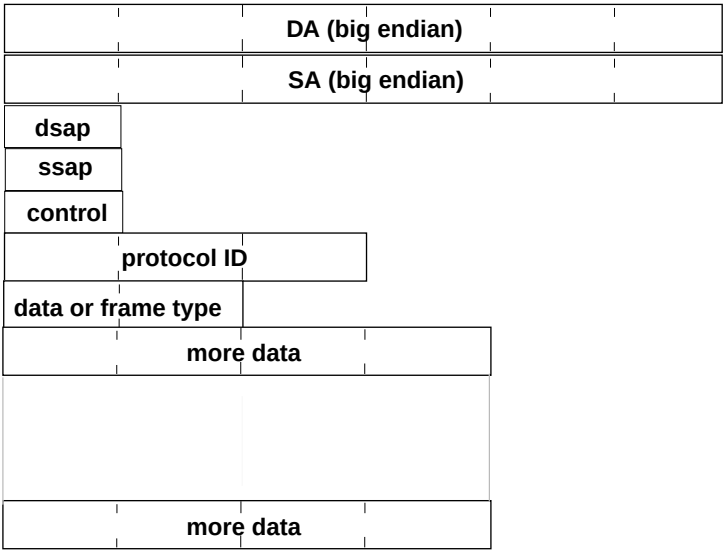


Figure B-2. Canonical Packet Format

IP version	header length	service type	total length	
identification		flags	fragment offset	
TTL	protocol	checksum		
		source IP address		
		destination IP address		
IP options (if any)...			padding (if necessary)	

Figure B-3. IP Header (After Canonical Packet Format)

UDP source port	UDP destination port
UDP message length	UDP checksum

Figure B-4. UDP Header (After IP Header)

source port		destination port	
sequence number			
acknowledgment number			
header length	reserved plus code bits		window
options (if any)...			padding (if necessary)

Figure B-5. TCP Header (After IP Header)

APPENDIX C

NULL MODEM CABLE PINOUTS

To connect LCM you need to insert a null modem cable at either the terminal end or the ATX port end. The null modem cable can be either a female DB25 or DB9 straight-through serial cable. Pinout information is provided in Figure C-1.

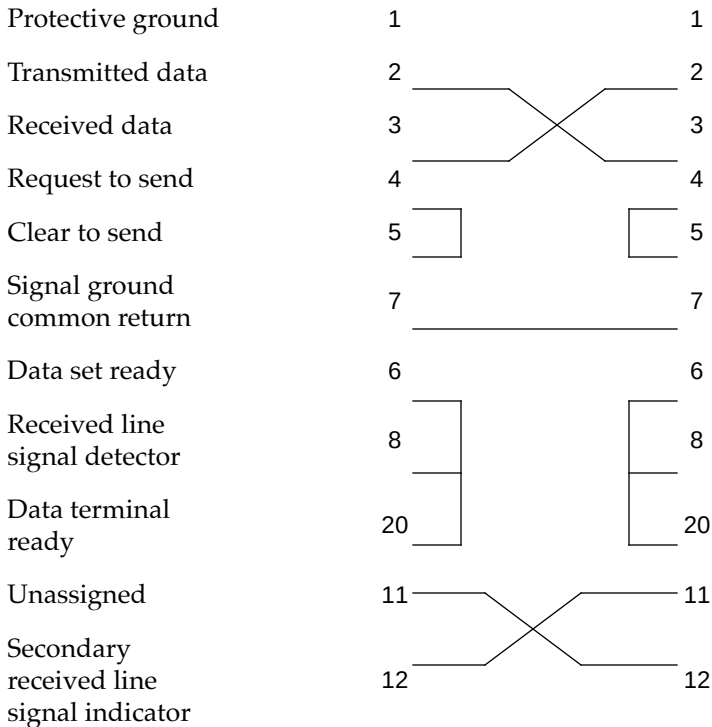


Figure C-1. Connector Pin Configuration For DB25 Null Modem Cable

APPENDIX D

GLOSSARY

4B/5B

Primary data encoding scheme used for FDDI.

AARP (AppleTalk Address Resolution Protocol)

AppleTalk ARP performs network address to datalink address mapping on Ethernet, Token Ring, and FDDI ports. This facility is similar to IP ARP with the additional capability to probe for active addresses as described in the address acquisition section.

address

A set of characters that uniquely identifies a station, peripheral device, node, or other unit in a network.

address table

A database of device addresses and their associated ports maintained by a bridge for use in making forwarding and filtering decisions.

address table filter

A mechanism for selectively forwarding or discarding (filtering) data that uses address table information to perform relatively simple filtering operations.

AEP (AppleTalk Echo Protocol)

Similar to IP ECHO (ping). This implementation will respond to, but not generate requests.

agent

Network management software that runs within a managed network device.

alarm

See *trap*.

ANSI

American National Standards Institute – One of several organizations that establishes standards which apply to internetworking and bridging.

ARP

address resolution protocol – An auxiliary protocol of the IP layer used to perform dynamic address translation between MAC addresses and internet addresses. Converts IP addresses to MAC addresses.

asynchronous transmission

Transmission of data according to token-holding rules. The station that holds the token may initiate data transmission if the token holding timer (THT) has not expired.

ATP (AppleTalk Transaction Protocol)

Used to provide reliable end-to-end transmission. This implementation provides only the “at least once” responder capability required for some ZIP operations.

attenuation

The amount of power (or light) lost as power travels through a medium, from the transmitter to the receiver. Difference between transmitted and received power, in decibels (dB).

AUI (attachment unit interface)

A standard connector type used for Ethernet connections.

backbone

The major, central transmission path for a network. A backbone usually handles high-volume, high-density traffic. Typically a backbone connects various LANs into an integrated network.

bandwidth

A measure of the amount of traffic a given medium can handle at one time: The communications capacity (measured in bits per second) of a transmission line or of a specific path through a network. Greater bandwidth generally means more information can be sent through a circuit during any given period of time.

beacon

A specialized frame that notifies stations on an FDDI ring that the ring is broken.

beacon process

Process by which stations on an FDDI network use a beacon frame to isolate a serious ring failure such as a break on the ring.

BPDU (bridge protocol data unit)

A data unit transmitted as part of the IEEE 802.1d Spanning Tree Protocol. The exchange of BPDUs allows bridges within a network to logically configure the network as a single spanning tree.

bps (bits per second)

The basic unit of data communications rate measurement.

bridge

An intelligent, protocol independent device used to connect similar or dissimilar LANs.

bursty

Adjective used to describe sporadic heavy volumes of network traffic (e.g., bursty traffic).

bypass

Optical or electronic isolation of a station from the network. A bypass situation typically occurs as a result of a station failure; the bypass allows the network to function normally, except for the absence of the failed station.

claim process

A process used to determine which station will initialize an FDDI ring.

claim token

Token issued by an FDDI station to alert other stations that it is about to initialize the ring.

combination port filter

A filter which may include several configurable fields and may be used to filter bridge traffic in a very specific manner.

concentrator

A device that provides attachment points for stations that are not connected directly to an FDDI dual ring. The concentrator is connected directly to the network; the stations connect to the concentrator.

congestion

A condition where a portion of the network is overloaded with more data than can be transmitted in the desired time period.

counter-rotating ring

Refers to two signal paths in a ring topology that send signals in opposite directions.

CSMA/CD (carrier-sense multiple access with collision detect)

A channel access (contention) method which requires each station to wait for an idle channel before transmitting. In addition, stations are able to detect overlapping transmissions (collisions) and retransmit in the event of a data collision.

DAC (dual attachment concentrator)

A concentrator connected to both the primary and secondary rings of an FDDI ring. A concentrator includes additional ports for the connection of other FDDI stations or concentrators.

DAS (dual attachment station)

An FDDI station connected to both the primary and secondary rings.

data link layer

Layer 2 in the OSI model. Defines frame construction, addressing, error detection and other services to higher layers.

datagram

Abbreviated and connectionless single-packet message sent from one station to another.

data rate (or speed)

The maximum number of bits of information that can be transmitted per second.

DDP (Datagram Delivery Protocol)

DDP is the network layer protocol for AppleTalk used for end-to-end addressing. The 13 byte header contains checksum, length, source and destination address, and packet type. The data portion of the packet is 0-586 bytes long.

destination address filtering

A process which discards (filters) local traffic and forwards only data intended for nodes on the extended LAN (non-local traffic).

downstream

Refers to the relative position of a station in a ring or network to another station in the same ring or network. A station is

downstream from another station if it receives the token or data after the other station receives the token or data.

dual homing

A method of connecting concentrators and stations that permits an alternate or backup path to the dual ring in case the primary connection fails.

dynamic address

An address “learned” by the bridge as opposed to addresses that are manually entered into the bridge's address table. The bridge “learns” addresses by reading them from the data packets it processes.

EIA (Electronic Industries Association)

Organization that sets standards for electrical interfaces (connectors).

encapsulation

A method for moving messages across networks that use different types of protocols. The message is encapsulated (rather than translated) so it can move across a network that otherwise could not understand its protocol. Encapsulating bridges generally use proprietary encapsulation schemes.

encode

To translate data into a series of electrical or optical pulses that can travel efficiently over a cable or other medium.

entity

An active element within an Open Systems Interconnection (OSI) network layer or sublayer.

Ethernet input/output module

The ATX component which accepts and sends data packets to and from a connected Ethernet network.

extended LAN

A collection of LANs interconnected by protocol-independent bridges.

FDDI (Fiber Distributed Data Interface)

A high-speed (100 Mb/s) access interface that uses two fiber optic cable rings operating in opposite directions to connect computers and peripheral equipment in a timed-token passing, dual ring of trees configuration. The dual ring architecture allows the network to continue operating if a station or cable fails.

FDDI input/output module

The ATX component which accepts and sends data packets to and from the connected FDDI network.

fiber optics

A technique that uses fiber optic transmitters, receivers and cables for the transmission of data.

filter

An instruction to a bridge to discard a certain type of data packets.

filtering rate

A measure (in packets per second) of a bridge's efficiency in examining each frame, comparing it with an address table, and then deciding whether to discard the frame or forward it.

forwarding rate

The rate (in packets per second) at which a bridge can receive a stream of packets from one network segment, complete all processing, and transmit the packets to another network segment.

fragment

Pieces of a frame left on an FDDI ring after a station has stripped the frame from the ring.

fragmentation

A process for breaking up large frames into smaller frames so they can be forwarded from one network to another (where the receiving network cannot accept the large frame size).

frame

A data message that includes a source address, destination address, data, frame check sequence (FCS) and control information.

full-wire speed

Refers to packet forwarding at the maximum rate at which data can be transmitted on a given LAN.

ICMP (Internet control message protocol)

An auxiliary protocol of IP used to convey advice and error messages about events in the IP layer.

IEEE (Institute of Electrical and Electronic Engineers)

International professional society which issues networking and other standards. The IEEE created the 802 family of LAN standards.

IEEE 802.2

The data link layer standard; used with IEEE 802.3, 802.4, and 802.5.

IEEE 802.3

The physical layer standard that uses the CSMA/CD access method on a bus topology LAN.

IEEE 802.4

The physical layer standard that uses the token-passing access method on a bus topology LAN.

IEEE 802.5

The physical layer standard that uses the token-passing access method on a ring topology LAN.

IEEE 802.6

Standard for metropolitan area networks (MANs) currently under development.

initialization

Transition of a device or network from startup state to operational state.

intelligent bridge

A bridge that is able to identify source and destination addresses.

internet

A large communications infrastructure composed of wide and local area networks. A generic reference to a network built using internetworking technology.

Internet

A large collection of connected networks which use TCP/IP. (Also referred to as the DARPA Internet, NSF/DARPA Internet or the Federal Research Internet.)

internetworking

The linking of one or more networks to facilitate communication across the networks.

interoperability

The ability of equipment from multiple vendors to exchange information using standardized protocols.

input-output module

ATX component which accepts and sends data packets to and from a connected network. Input-output modules include the Ethernet modules, the FDDI modules and the Token Ring concentrator

module.

I/O

See input-output module.

IP (Internet protocol)

IP is the basic datagram protocol used at the network layer of the TCP/IP stack.

ISO (International Standards Organization)

An organization that creates, controls and publishes standards.

jitter

Clocking deviation on a network.

Kbps (kilobits per second)

1,000 bits per second.

LAN (local area network)

A network that interconnects a variety of devices (computers, printers, servers, etc.) within a limited geographical area. A LAN typically connects devices within a building or campus.

link-loss budget

Each connection (link) in an optical system results in a certain amount of signal strength loss. Link-loss budget refers to the process of calculating link loss for the entire system. If the total link loss exceeds a certain limit, the system will not function.

LLC (logical link control)

A part of the data link layer of the OSI model that defines the transmission of a frame of data between two stations (with no intermediate switching nodes).

LMA (local management agent)

Software running on a network device to control the device in terms of network management functions.

local traffic

Traffic within a given network segment.

logical ring

The circular path a token follows in an FDDI network. The logical ring may be considered to be circular, although the actual physical ring may be a ring, a tree, or a dual ring of trees.

MAC (media access control)

The data link layer sublayer responsible for scheduling, transmitting and receiving data on a shared medium local area network.

mask

A subset of a larger set of data to be included for comparison, analysis, etc. For example, in bridge filtering, a mask might be specified to include only the first four address bits as the basis for filtering decisions.

Mbps (megabits per second)

1 million bits per second.

MIB (management information base)

A collection of objects unique to a specific device that can be accessed via a network management protocol. The ATX has its own MIB.

MIC (media interface connector)

Optical fiber connector type used for ATX bridge FDDI port. A MIC consists of two parts: the MIC plug, which terminates the optical fiber cable, and the MIC receptacle on the FDDI node or station.

multicast

Packets destined for more than one address.

multicast storm

Excessive multicast packet traffic, typically generated by a faulty device. Multicast storms can cause severe network performance problems.

NBP (Name Binding Protocol)

In the AppleTalk routing protocol, name binding is the process of mapping a name (object:type@zone tuple) to a socket address. Instead of maintaining a database, name binding works by flooding queries on all networks that have the specified zone. The first step in this process is an end-node sending a broadcast request to a router. The router then determines all the networks with the specified zone and addresses a forward request to the closest

router on each of those networks. The destination router(s) then multicasts a lookup request on the destination network. A response is then returned by an end-node in a directly addressed packet. Note that wildcards are allowed to enable the chooser to display all objects of any given type.

network

Interconnected computer systems, terminals, and data communication facilities. A network must have at least three endpoints and may have any number of links and nodes.

node

Any device connected to a communication network. Examples: computer, workstation, printer, server, concentrator, bridge.

NZRI (nonreturn to zero inverted)

Secondary data encoding scheme used by FDDI; used to enhance the reliability of 4B/5B encoding. (Same as NRZM, nonreturn to zero-mark.)

OBS (optical bypass switch)

A switch that uses an optical mechanism to automatically bypass a malfunctioning or powered-off station on an FDDI network. Prevents unnecessary ring initialization.

optical receiver

A circuit that converts an incoming optical signal to an electrical signal.

optical transmitter

A circuit that converts an electrical signal to an optical signal.

OSI (Open Systems Interconnection)

Refers to the OSI reference model, a logical structure for network operations. OSI is the internationally accepted framework of standards for internetwork communication.

packet

A group of bits including data and control elements arranged in a specific format that are transmitted and switched as a composite whole. Control elements include a source address, destination address, frame control and status indicators and a frame check sequence (FCS).

packet processing engine module

High-performance component of the ATX bridge capable of performing packet analysis / transfer at a high rate.

PDU (protocol data unit)

The portion of a datagram that contains the data associated with a particular protocol.

peer-to-peer

Term used to describe data transmission between entities in the same sublayer of the OSI model.

PHY (Physical Layer Protocol)

FDDI standard that defines symbols, line states, clocking

requirements and the encoding of data for transmission.

physical layer

Layer 1 of the OSI model. Defines and handles the electrical and physical connections between systems.

PMD (Physical Layer Medium Dependent)

FDDI standard that defines the medium and protocols used to transfer symbols between physical layer protocols.

power budget

The difference between transmit power and receiver sensitivity, including any safety margins.

PPP (point-to-point protocol)

A protocol for transmitting datagrams (IP or MAC packets) over a serial point-to-point link (e.g., the out-of-band management port).

pps (packets per second)

Unit of measure used to express packet data throughput. 18 pps is approximately equal to 9600 bps.

propagation delay

The time it takes for a signal to travel across a network.

protocol

A set of rules used by computers and related devices to communicate with each other.

protocol suite

A group of protocols related to a common framework.

RARP (reverse address resolution protocol)

A protocol that translates MAC addresses to IP addresses.

ring

A network of stations that uses a circular logical topology. Data is passed from station to station, for examination or copying, and is finally returned to the originating station, which removes the data it transmitted from the network.

RISC (Reduced Instruction Set Computing)

A data processing technology in which functions are performed using the least possible number of instructions to yield very fast processing.

RTMP (Routing Table Maintenance Protocol)

In the AppleTalk routing protocol, RTMP is a distance vector routing protocol similar to RIP. It keeps track of hop count/destination network range pairs. Split Horizon is performed, but poison reverse and triggering are not. Periodic updates are sent once every 10 seconds. Individual route lifetimes are not kept. Instead, all routes are marked as “suspect” once every 20 seconds. If the route is not updated by the next 20 second interval, it is marked bad. Following two or more intervals, the route is removed from the table.

SAC (single attachment concentrator)

A concentrator with a slave (S) port for attachment to the FDDI

network and master (M) ports for the attachment of stations or other concentrators.

SAS (single attachment station)

An FDDI station that uses only one connection (an S port) for connection to the FDDI ring.

segment

When two or more networks are interconnected to form an internetwork, the original networks are referred to as segments.

service

A set of functions offered to a user by a provider.

SMT (station management)

Refers to the entity within a station on an FDDI ring that monitors and controls station activity.

SNMP (simple network management protocol)

A TCP/IP protocol for communication between a network management system and a network device.

source address filtering

A bridge function that forwards or rejects data, depending on the data's source address.

static address

Addresses manually entered into the bridge's address table (as

opposed to those automatically “learned” by the bridge).

STP (spanning tree protocol)

A protocol which ensures that only one path will be used between two devices; prevents active loops (multiple paths to devices) by closing certain paths. With STP operating, a redundant link serves as a backup link only if a normal path fails.

symbol

The smallest signaling element used by the MAC sublayer. Each symbol corresponds to a specific sequence of code bits to be transmitted by the physical layer.

synchronous transmission

A transmission technique in which an uninterrupted block of data is transmitted, using no redundant information such as stop and start bits to identify the beginning and end of a unit of data.

TCP/IP (transmission control protocol/Internet protocol)

Internetworking protocols sometimes referred to as the Internet suite of protocols.

THT (token holding timer)

A timer that controls the amount of time a station on a token passing network may hold the token in order to transmit data.

token

A unique symbols that circulates around a network ring following a data transmission. The token lets stations know when they can

transmit.

token ring

Local area network access mechanism and topology in which a supervisory frame (the token) is passed from station to station. Stations wishing to gain access to the network wait for the token to arrive before transmitting data.

topology

The arrangement of devices and cables that make up a network.

translating bridge

A bridge that can pass data between LANs that use different protocols.

translation

Modification of data packets from one type of network so they can be used on a different type of network (e.g., Ethernet to FDDI translation).

trap

Alarm; notification of an event that has occurred on a network. Some alarms require intervention or action by the network administrator; some are merely informational.

TRT (token rotation timer)

A mechanism by which a station on an FDDI ring measures the time since it last received a token frame.

TTRT (target token rotation time)

A time defined for tokens to travel around an FDDI ring; used to synchronize the clocking of traffic on the ring.

UDP (user datagram protocol)

A TCP/IP protocol for the connectionless transport layer.

upstream

Refers to the relative position of a station in a ring or network to another station in the same ring or network. A station is upstream from its neighbor if it receives the token or data before its neighbor receives the token or data.

virtual LANs

This implementation is capable of creating virtual LANs, i.e. logical AppleTalk subnets that span more than one port and, by extension, more than one router. A virtual LAN has a single network range and zone list. Traffic arriving on one physical network and destined for a different physical network on the same VLAN is bridged between ports. Traffic arriving on one VLAN and destined to another is routed with the output port being determined via AARP

The address acquisition process raises the complexity relative to non-AppleTalk protocols due to the fact that acquisition must be performed on all ports of the VLAN simultaneously. Internal to the routing engine this is accomplished using a single circuit which is mapped via a mask to physical ports. Externally, this means that modifications to the parameters of one VLAN port are applied to all ports in the VLAN.

A VLAN is defined by setting the network range of multiple ports to the same value with the exception that ports configured with range 0-0 are on different virtual segments. Virtual LAN port

groups must consist of ports with all the same underlying link type.

WAN (wide area network)

A communication network that spans a large geographic area.

ZIP (Zone Information Protocol)

In the AppleTalk routing protocol, ZIP is used to disseminate zone information from routers to end nodes and between routers. End nodes ask for their zone on start-up and a global list of available zones each time someone opens the Chooser. Routers bind a zone list to a network number when a network number is learned via RTMP. This is done by querying the router from which the route was learned.

Note that there is no way to change the network.zone list binding after a binding is performed. The only way to add or delete a zone from a network is to isolate the network from the internet and then have the internet relearn the route. Since RTMP does not have a triggered update, ensuring a network has been deleted can take quite a while; up to ten minutes according to the specification.

APPENDIX E

BIG ENDIAN TO LITTLE ENDIAN CONVERSION

The chart below provides the bit swap values and a conversion formula.

Table E-1. Big Endian To Little Endian Conversion Chart

Big Endian value	Big Endian bits	Little Endian bits	Little Endian value
0	0000	0000	0
1	0001	1000	8
2	0010	0100	4
3	0011	1100	C
4	0100	0010	2
5	0101	1010	A
6	0110	0110	6
7	0111	1110	E
8	1000	0001	1
9	1001	1001	9
A	1010	0101	5
B	1011	1101	D
C	1100	0011	3
D	1101	1011	B
E	1110	0111	7
F	1111	1111	F

The conversion process has two steps, first you swap the bits, then you use the conversion chart above to convert the swapped bits to the little endian format.

1. First, swap the big endian bits, use the conversion chart to find the equivalent values. For example:

00 00 F6 09 47 88

00 00 6F 90 74 88

2. Now that you have the bits swapped, use the conversion chart to find the equivalent values. For example:

0 = 0

6 = 6

F = F

9 = 9

7 = E

4 = 2

8 = 1

So the little endian equivalent is 00 00 6F 90 E2 11.

INDEX

A

adding

- filters 5-15

- IP addresses 3-6

- IPX addresses 3-13

address classes, IP 3-5

Address Resolution Protocol. See
ARP

address table filters

- about 5-4

- combination address 5-6

- destination address 5-5

- source address 5-5

- source address multicast 5-6

addresses

adding

- IP 3-6

- IPX 3-13

- changing, IPX 3-13

deleting

- IP 3-6

displaying

- IP 3-7

- IPX 3-13

aging time, defined 3-25

applications of, filters 5-19

ARP, defined 1-22

assigning

- community name 4-20

- IP addresses 3-6

- IPX addresses 3-13

authentication password, defined 3-
24

B

basic LCM commands 1-41

baud rate

- displaying 4-20

- setting 4-20

Bridging

- method 1-14

- source routing 1-17

- SRT 1-19

- transparent 1-16

bridging functions

- about 3-1

- defined 3-3

- disabling 3-5

- displaying 3-4

- enabling 3-3

- overview 1-13

C

canonical format B-1

changing

- IPX addresses 3-13

- subnet mask 3-7

checksum comparison test 7-4

cleaning fan filters 8-4

combination port filters

- about 5-7

- configurable fields 5-8

community name, assigning 4-20

configuration alarm, defined 3-26

configuring

- bridging, about 3-1

- IP routing 3-5

- IPX routing 3-12

connecting

- LCM 2-10

connectivity problems,

- troubleshooting 7-9

conventions, LCM command 1-40

D

deleting

- filters 5-18

- IP addresses 3-6

diagnostics

- checksum comparison 7-4

- operational 7-5

- overview 7-1

- power-up 2-6, 7-2

- results 7-6

disabling

- bridging functions 3-5
- IP routing 3-12
- IPX routing 3-15
- ports 4-18
- displaying
 - baud rate 4-20
 - bridge functions 3-4
 - ES/1 status 4-13
 - filters 5-19
 - IP addresses 3-7
 - IP routing functions 3-12
 - IPX addresses 3-13
 - IPX routing functions 3-15
 - MAC addresses 4-15
 - manufacturing information 4-17

E

- enabling
 - bridging functions 3-3
 - IP routing 3-8
 - IPX routing 3-14
 - modules 4-19
 - ports 4-18
- ES/1
 - mounting 2-3
- ES/1 diagnostics 7-1
- examples of filters 5-19

F

- failure indicators 7-5
- fan filters, cleaning 8-4
- filters
 - adding 5-15
 - address table
 - about 5-4
 - combination address 5-6
 - destination address 5-5
 - source address 5-5
 - source address multicast 5-6
 - combination port
 - about 5-7

- configurable fields 5-8
- deleting 5-18
- displaying 5-19
- enhancing performance 5-3
- examples of 5-19
- firewall, example 5-27
- modifying 5-18
- performance considerations 5-2
- security example 5-20
- security uses 5-2
- type field defined 5-9
- firewall filters, example of 5-27
- fuses, replacing 8-3

G

- get password, defined 3-25

I

- installing modules 8-1
- IP addresses
 - about 3-5
 - assigning 3-6
 - deleting 3-6
 - displaying 3-7
- IP routing
 - disabling 3-12
 - displaying functions 3-12
 - enabling 3-8
 - functions defined 3-8
 - overview 1-22
- IP subnet mask, changing 3-7
- IPX addresses
 - assigning 3-13
 - changing 3-13
 - displaying 3-13
- IPX routing
 - disabling 3-15
 - displaying 3-15
 - enabling 3-14
 - overview 1-32

L**LCM**

- connecting 2-10
- description of 1-39

LCM command syntax 1-40

LED sequence

- normal operation 7-6
- power-up 7-2

LEDs, front panel meaning 2-2

Local Console Manager. See
LCM 1-39

loopback tests 7-5

M

MAC addresses, displaying 4-15

manufacturing information,
displaying 4-17

MIB variables, modifying 3-23
modifying

- filters 5-18
- MIB variables 3-23

module types, defined 4-14

modules

- installing 8-1
- restarting 4-19
- stopping 4-19
- swapping 8-2

monitoring statistics 4-1

mounting ES/1 2-3

multicast storm protection

- defined 3-22
- MIB variables 3-22

N

NMS failure traps 7-5

null modem cable pinouts C-1

O

operational diagnostics 7-5

- loopback tests 7-5

P

packet conversion format B-1

performance, enhancing with
filters 5-3

pinouts, null modem cable C-1
ports

- disabling 4-18
- enabling 4-18

power-up

- LED sequence 2-7, 7-2
- troubleshooting 2-7

power-up diagnostics 2-6, 7-2

- failure indicators 7-5
- NMS failure traps 7-5
- results 7-4
- specific tests 7-3

Proxy ARP, defined 1-23

R

replacing fuses 8-3

restarting modules 4-19

RIP, defined 1-22

routing functions

- disabling
 - IP 3-12
 - IPX 3-15

- displaying
 - IP 3-12
 - IPX 3-15

- enabling
 - IP 3-8
 - IPX 3-14

IP defined 3-8

overview

- IP 1-22
- IPX 1-32

Routing Information Protocol. See
RIP

S

SAP, defined 1-33

Service Advertising Protocol. See

SAP

set password, defined 3-24

setting baud rate 4-20

statistics, monitoring 4-1

status, displaying

ES/1 4-13

LED 7-6

stopping modules 4-19

subnet mask, IP, changing 3-7

swapping modules 8-2

switches, front panel, meaning 2-3

syntax, LCM command 1-40

system contact, defined 3-23

system location, defined 3-24

system name, defined 3-24

T

taking offline, modules 4-19

translating bridge, overview 1-20

trap PDUs, UDP port number 6-1

traps (acknowledge), defined 3-25

troubleshooting power-up 2-7, 7-9

type, filter 5-9